

**EVOLUTION STRATEGIES UNTUK OPTIMASI PEMBENTUKAN
FUNGSI REGRESI LINIER DALAM MENENTUKAN
KEBUTUHAN VOLUME AIR PENYIRAMAN TANAH**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

EVOLUTION STRATEGIES UNTUK OPTIMASI PEMBENTUKAN FUNGSI REGRESI LINIER DALAM
MENENTUKAN KEBUTUHAN VOLUME AIR PENYIRAMAN TANAH

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

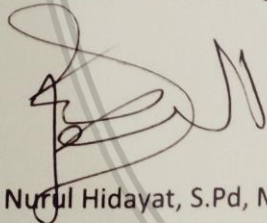
Robertus Santoso Aji Putra

NIM: 125150205111009

Skrripsi ini telah diuji dan dinyatakan lulus pada
03 Agustus 2018

Telah diperiksa dan disetujui oleh:

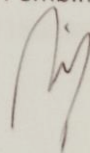
Dosen Pembimbing I



Nurul Hidayat, S.Pd, M.Sc

NIP: 19680430 200212 1 001

Dosen Pembimbing II



M. Ali Fauzi, S.Kom, M.Kom

NIK: 201502 890101 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Asstoto Kurniawan, S.T, M.T, Ph.D

NIP: 19710518 200312 1 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 03 Agustus 2018



Robertus Santoso Aji Putra

NIM: 125150205111009

KATA PENGANTAR

Segala puji dan hormat syukur penulis panjatkan kepada Allah Bapa Yang Maha Kuasa atas rahmat, berkat dan karuniaNya sehingga penulis dapat menyelesaikan skripsi yang berjudul “EVOLUTION STRATEGIES UNTUK OPTIMASI PEMBENTUKAN FUNGSI REGRESI LINIER DALAM MENENTUKAN KEBUTUHAN VOLUME AIR PENYIRAMAN TANAH”.

Selama proses penyusunan skripsi ini, penulis mendapat banyak bimbingan, kritik, saran, dukungan, motivasi, maupun doa dari berbagai pihak. Maka dari itu, penulis mengucapkan penghargaan dan rasa terimakasih kepada:

1. Nurul Hidayat, S.Pd, M.Sc dan M. Ali Fauzi, S.Kom, M.Kom selaku Dosen Pembimbing Skripsi yang telah memberikan bimbingan dan arahannya selama proses penyusunan skripsi ini sehingga penulis dapat menyelesaikannya dengan baik dan lancar.
2. Ayah dan Ibu penulis yang selalu memberikan dukungan baik secara moral dan materi, mendoakan serta memberi semangat kepada penulis sehingga dapat menyelesaikan dengan lancar.
3. Sahabat gang sebuku yang telah bersedia meluangkan waktu, pikiran, dan tenaganya untuk bersedia membantu dan menyemangati penulis.
4. Saudara/I OMK Dewandaru dan Keluarga Besar KMK St. Isidorus FILKOM
5. Teman-teman seperjuangan skripsi untuk semua dukungan dan semangatnya.
6. Dan semua pihak yang tidak bisa penulis sebutkan satu per satu.
Terimakasih atas segala bantuannya.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan dan masih belum sempurna, karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Maka, kritik dan saran yang membangun dari semua pihak sangat diharapkan demi penyempurnaan selanjutnya. Semoga skripsi ini dapat bermanfaat dan berguna bagi semua pihak, baik penulis maupun pembaca. Terima kasih.

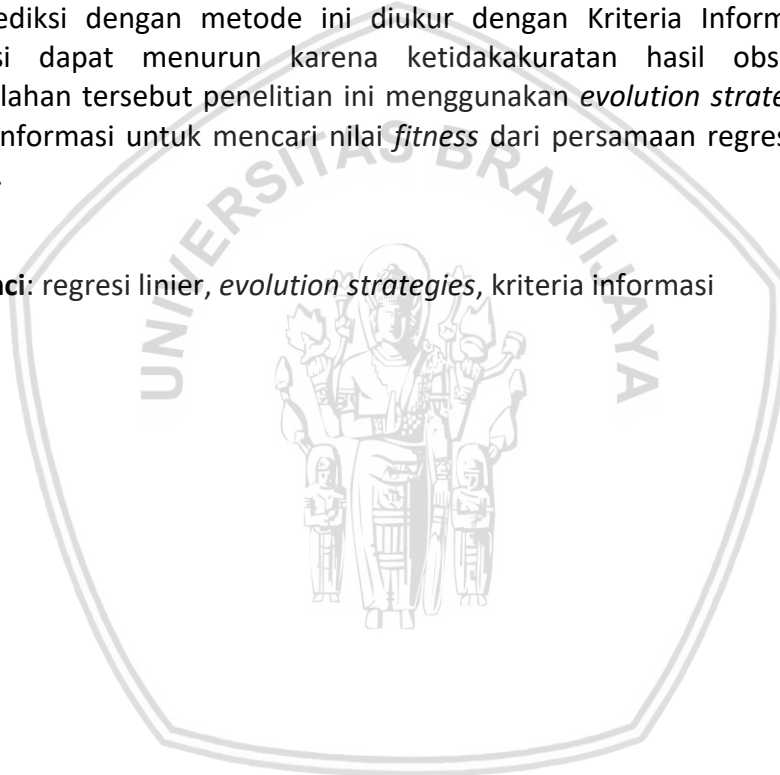
Malang, 03 Agustus 2018

Robertus Santoso Aji Putra
ajiputra001@gmail.com

ABSTRAK

Laboratorium Benih Balai Pengkajian Teknologi Pertanian Jawa Timur adalah salah satu unit kerja milik pemerintah provinsi yang ditunjuk sebagai pelaksana teknis untuk melakukan pengkajian daya tumbuh benih. Saat ini di tempat tersebut sedang dikembangkan alat siram otomatis berdasarkan sensor kelembapan tanah, namun alat itu belum bisa memprediksi kebutuhan volume air guna menjaga kelembapan media penumbuhan benih. Dengan bantuan sensor kelembapan pada alat dan pengetahuan pakar, data berupa kumpulan amatan kelembapan tanah terhadap kebutuhan volume air telah didapatkan. Penelitian ini dilakukan untuk menerapkan metode regresi linier agar alat tersebut dapat melakukan prediksi berdasarkan pola data dalam bentuk persamaan. Ketepatan hasil prediksi dengan metode ini diukur dengan Kriteria Informasi. Kriteria Informasi dapat menurun karena ketidakakuratan hasil observasi. Dari permasalahan tersebut penelitian ini menggunakan *evolution strategies* dengan kriteria informasi untuk mencari nilai *fitness* dari persamaan regresi yang telah dihitung.

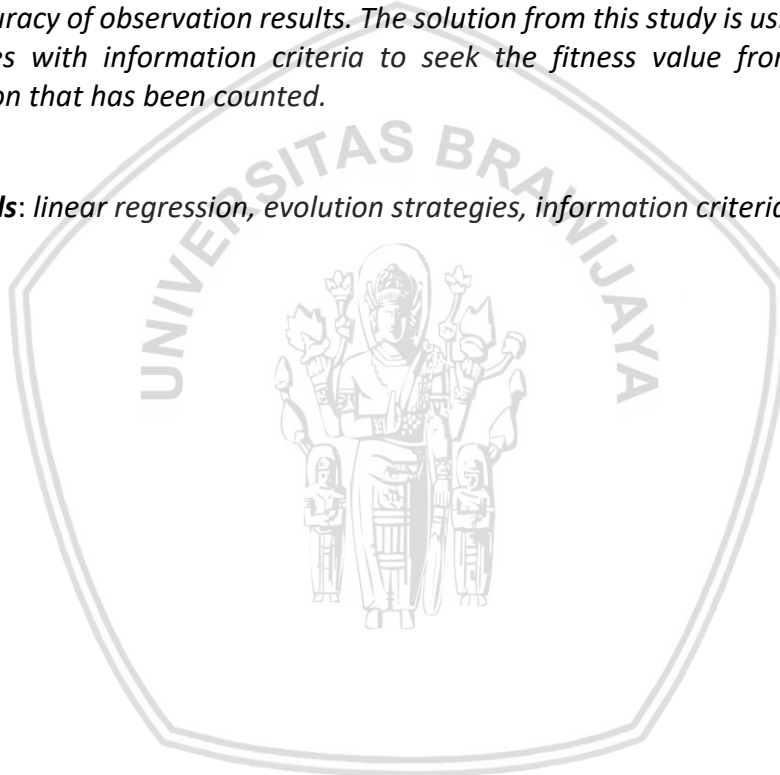
Kata kunci: regresi linier, *evolution strategies*, kriteria informasi



ABSTRACT

Seed Laboratory BPTP East Java is one of provincial government work units that have assignment as the technical implementer to conduct a study in seed growth. Currently at this place is being developed automatic watering device based on soil humidity sensor, but the device cannot predict the volume of water needed in order to keep the moist of seed growth media. With the help of humidity sensor on device and expert's knowledge, the observations dataset of soil moisture to the needs of waters volume has been obtained. This study was conducted to apply linear regression method so that the device can perform predictions based on dataset patterns as an equation. The accuracy of prediction results with this method is measured by the information criteria. Information criteria can be decreasing due to Inaccuracy of observation results. The solution from this study is using evolution strategies with information criteria to seek the fitness value from the linear regression that has been counted.

Keywords: *linear regression, evolution strategies, information criteria*



DAFTAR ISI

ABSTRAK.....	2
ABSTRACT	6
DAFTAR ISI	7
BAB 1 PENDAHULUAN.....	11
1.1 Latar belakang.....	11
1.2 Rumusan masalah.....	13
1.3 Tujuan	13
1.3.1 Tujuan khusus	13
1.4 Manfaat.....	13
1.4.1 Bagi Peneliti.....	14
1.4.2 Bagi Instansi	14
1.5 Batasan masalah	14
1.6 Sistematika pembahasan.....	14
1.6.1 BAB I Pendahuluan.....	14
1.6.2 BAB II Landasan Pustaka	14
1.6.3 BAB III Metodologi	15
1.6.4 BAB IV Perancangan.....	15
1.6.5 BAB V Implementasi.....	15
1.6.6 BAB VI Pengujian dan Analisis.....	15
1.6.7 BAB VII Penutup	15
BAB 2 LANDASAN PUSTAKA	16
2.1 Kajian Pustaka	16
2.2 Regresi Linier.....	17
2.3 Optimasi Regresi Linier	18
2.3.1 Sisaan (<i>Error</i>)	18
2.3.2 Kriteria Informasi	19
2.4 Evolution strategies	20
2.4.1 Struktur Dasar Algoritma <i>Evolution strategies</i>	21

2.4.2 Representasi Kromosom	24
2.4.3 Inisialisasi	25
2.4.4 Representasi.....	25
2.4.5 Evaluasi.....	26
2.4.6 Seleksi.....	27
2.5 Pengujian Regresi Linier.....	Error! Bookmark not defined.
BAB 3 METODOLOGI	28
3.1 Studi Pustaka	29
3.2 Pengumpulan Data	29
3.3 Analisis Kebutuhan	32
3.4 Perancangan Sistem.....	33
3.4.1 Deskripsi Umum Sistem	33
3.4.2 Arsitektur Sistem.....	33
3.5 Implementasi Sistem	33
3.6 Pengujian dan Analisis	34
3.7 Kesimpulan dan Saran	34
BAB 4 PERANCANGAN.....	35
4.1 Deskripsi Umum Sistem.....	35
4.2 Batasan Sistem.....	35
4.3 Alur Kerja Sistem.....	35
4.3.1 Komponen Regresi Linier	37
4.3.2 <i>Evolution strategies</i>	42
4.4 Perhitungan Manual	45
4.4.1 Proses Perhitungan Nilai Kriteria Informasi.....	Error! Bookmark not defined.
4.5 Perancangan Tampilan Antarmuka	52
4.6 Perancangan Pengujian dan Analisis	53
4.6.1 Perancangan Pengujian Populasi	53
4.6.2 Perancangan Pengujian <i>Offspring</i>	54
4.6.3 Perancangan Pengujian Generasi	54
BAB 5 IMPLEMENTASI	56

5.1 Perangkat Implementasi	56
5.2 Implementasi Proses Regresi Linier	56
5.2.1 Implementasi Pembentukan Persamaan Regresi	62
5.2.2 Implementasi Pembentukan Sisaan	63
5.2.3 Implementasi Perhitungan Nilai Kriteria Informasi	64
5.3 Implementasi Proses <i>Evolution strategies</i>	65
5.3.1 Implementasi Inisialisasi Populasi	66
5.3.2 Implementasi Perhitungan Generasi	67
5.4 Implementasi Tampilan Antarmuka	68
BAB 6 PENGUJIAN DAN ANALISIS	56
6.1 Pengujian Parameter <i>Evolution strategies</i>	73
6.1.1 Pengujian Jumlah Populasi	73
6.1.2 Pengujian Jumlah <i>Offspring</i>	74
6.1.3 Pengujian Jumlah Generasi	76
BAB 7 PENUTUP	78
7.1 Kesimpulan	78
7.2 Saran	78
DAFTAR PUSTAKA	79



BAB 1 PENDAHULUAN

Bab ini berisi penjelasan tentang permasalahan yang menjadi latar belakang penelitian. Pada latar belakang hal yang dibahas adalah permasalahan yang diangkat dalam penelitian dan juga tentang metode yang digunakan untuk menyelesaikan permasalahan tersebut. Bab ini juga akan membahas tentang masalah yang sudah dirumuskan dari latar belakang serta tujuan, manfaat, batasan dan sistematika pembahasan.

1.1 Latar belakang

Balai Pengkajian Teknologi Pertanian Jawa Timur yang selanjutnya disingkat BPTP Jatim adalah unit kerja milik pemerintah provinsi yang memiliki salah satu visi yaitu menyediakan teknologi yang dapat menunjang komoditas pertanian sesuai dengan kondisi lingkungan yang dibutuhkan. Salah satu fasilitas yang dimiliki BPTP Jatim adalah Laboratorium Benih, yang ditunjuk sebagai pelaksana teknis untuk melakukan pengkajian kualitas benih. Tugas utama dari laboratorium ini adalah melayani permintaan analisis dari berbagai instansi untuk kepentingan penelitian serta melayani permintaan analisis langsung dari petani (Balai Pengkajian Teknologi Pertanian Jawa Timur, 2010).

Laboratorium Benih BPTP Jatim telah memiliki beberapa alat serta metode untuk menguji kualitas benih, namun pengujian kualitas benih harus disertai dengan pengujian daya tumbuh benih. Sampai saat ini penyiraman media penumbuhan benih di Laboratorium Benih BPTP Jatim masih menggunakan serangkaian pipa untuk menyalurkan air yang bekerja secara manual guna menghasilkan kelembapan tanah ideal. Hal ini tentu dapat mempengaruhi hasil pertumbuhan benih, dikarenakan untuk tumbuh secara optimal benih membutuhkan volume penyiraman air yang sesuai guna menjaga kelembapan tanah tetap ideal (FAO Land & Water Division, 2003). Diharapkan dengan adanya alat yang dapat menjaga kelembapan tanah secara otomatis di Laboratorium Benih BPTP Jatim hasil pengujian daya tumbuh benih lebih optimal, dikarenakan penyiraman media penumbuhan benih lebih tepat sesuai kebutuhan. Saat ini di Laboratorium Benih BPTP Jatim sedang dikembangkan alat yang dapat melakukan penyiraman secara otomatis. Alat yang sedang dikembangkan tersebut menggunakan sensor yang berfungsi sebagai penerima masukan tingkat kelembapan tanah. Hal ini bertujuan untuk menentukan volume air yang dibutuhkan dalam penyiraman, namun belum ada penghubung antara masukan dan keluaran pada alat tersebut. Tanpa adanya penghubung antara masukan dan keluaran pada alat yang handal, akan muncul masalah seperti jumlah penyiraman yang terlalu sering ataupun terlalu sedikit yang akan sangat mempengaruhi kandungan nutrisi tanah dan dapat mempengaruhi pertumbuhan tanaman. Oleh

karena itu diperlukanlah metode untuk menghubungkan antara input dan output pada alat.

Berdasarkan permasalahan di atas, peneliti ingin menerapkan suatu persamaan pada pengembangan alat siram otomatis yang sedang dikembangkan, sehingga alat tersebut dapat memprediksi kebutuhan volume air berdasarkan kelembapan tanah. Persamaan tersebut didapat dari metode yang dapat mengenali pola hubungan kebutuhan volume air berdasarkan kelembapan tanah awal hingga mencapai ideal. Dengan bantuan sensor kelembapan pada alat yang sedang dikembangkan dan pengetahuan pakar terhadap kelembapan tanah ideal, data berupa kumpulan amatan kebutuhan volume air berdasarkan kelembapan tanah dapat dicari. Data yang digunakan pada penelitian ini merupakan kumpulan amatan yang cenderung membentuk pola garis lurus dengan asumsi semakin kecil kelembapan tanah maka semakin besar volume air yang dibutuhkan atau sebaliknya. Metode yang cocok digunakan untuk permasalahan ini adalah regresi linier. Regresi linier mempunyai persamaan yang disebut sebagai persamaan regresi. Persamaan regresi merumuskan hubungan yang lurus antara satu amatan dengan amatan lainnya. Setiap amatan mempresentasikan hubungan variabel terikat dengan satu atau lebih variabel bebas (Levin & Rubin, 1998).

Dalam regresi linier, bisa jadi persamaan yang terbentuk kurang kuat untuk menggambarkan pola hubungan antar amatan. Persamaan regresi yang kurang kuat dapat menyebabkan prediksi menjadi salah. Hal ini dikarenakan adanya amatan pencilan (*outliers*) di dalam data yang ikut membentuk persamaan regresi. Amatan pencilan merupakan amatan yang memiliki nilai terlalu jauh dari amatan lainnya. Dalam penelitian ini amatan pencilan bisa jadi ada di dalam data karena ketidakakuratan sensor dalam menentukan kelembapan tanah atau kurangnya ketelitian dalam mengamati kebutuhan volume air. Untuk mengantisipasi kesalahan prediksi, perlu adanya pendeteksian amatan pencilan sebelum membentuk persamaan regresi (Johnson & Wichern, 1996). Pendeteksian amatan pencilan bisa melalui beberapa kali proses. Hal ini bisa terjadi karena setelah amatan yang diduga sebagai pencilan dihapus persamaan regresi yang terbentuk akan berubah. Perubahan persamaan regresi membuat amatan yang tadinya bukan pencilan menjadi pencilan baru. Selain dibutuhkan proses yang berulang, satu proses pendeteksian membutuhkan beberapa tahap yang harus dilalui (Hadi & Simonoff, 1993). Banyaknya tahap yang dilalui serta proses yang berulang membuat cara ini kurang efisien sehingga perlu satu cara untuk melakukan deteksi pencilan secara serempak. Pada tahun 1992 Mills dan Prasad mengusulkan metode pendeteksian amatan pencilan secara serempak dengan rumus kriteria informasi. Amatan pencilan ditentukan dengan menghitung nilai kriteria informasi terbaik berdasarkan kombinasi amatan yang dimasukkan dalam rumus. Kelemahan dari metode ini adalah semua kombinasi harus dicoba untuk menghasilkan nilai kriteria informasi terbaik sehingga mengurangi efisiensi. Solusi yang dapat digunakan untuk menyelesaikan permasalahan ini adalah *evolution*

strategies. *Evolution strategies* dapat mencari nilai kriteria informasi terbaik tanpa mencoba semua kombinasi (Alma, Kurt, & Ugur, 2008).

Berdasarkan pemaparan latar belakang di atas, maka peneliti ingin meneliti *Evolution strategies* untuk Optimasi Pembentukan Fungsi Regresi Linier dalam Menentukan Kebutuhan Volume Air Penyiraman Tanah. Hasil dari penelitian ini berupa persamaan yang akan digunakan pada pengembangan alat siram otomatis di Laboratorium Benih BPTP Jatim. Dengan menerapkan hasil penelitian ini pada pengembangan alat siram otomatis tersebut, diharapkan dapat membantu Laboratorium Benih BPTP Jatim dalam menjaga kelembapan media penumbuhan benih.

1.2 Rumusan masalah

Berdasarkan paparan latar belakang tersebut, maka rumusan masalah yang dapat dikaji adalah sebagai berikut:

1. Bagaimana menerapkan metode regresi linier guna membentuk persamaan regresi untuk menghitung volume air yang dibutuhkan berdasarkan kelembapan tanah?
2. Bagaimana menerapkan metode *evolution strategies* guna mengoptimasi metode regresi linear?
3. Bagaimana hasil pengujian variable pada *evolution strategies*?

1.3 Tujuan

Berdasarkan permasalahan yang telah dijelaskan, penelitian ini memiliki beberapa tujuan antara lain:

1. Menerapkan metode regresi linier guna membentuk persamaan regresi untuk menghitung volume air yang dibutuhkan berdasarkan kelembapan tanah.
2. Mengimplementasikan optimasi metode regresi linier menggunakan *evolution strategies*.
3. Melakukan pengujian terhadap variable yang terdapat pada *evolution strategies*.

1.3.1 Tujuan khusus

Membantu Laboratorium Benih BPTP Jatim dalam pengembangan alat siram otomatis untuk menjaga kelembapan media penumbuhan benih dengan memberikan metode yang tepat untuk mengolah hasil inputan dan memberikan keluaran yang sesuai.

1.4 Manfaat

Dengan adanya penelitian ini akan ada manfaat yang dirasakan oleh pihak peneliti dan instansi terkait, manfaat tersebut antara lain:

1.4.1 Bagi Peneliti

1. Sebagai media untuk menerapkan ilmu pengetahuan teknologi khususnya dibidang komputasi cerdas.
2. Mendapatkan pemahaman tentang penerapan metode regresi linier yang dioptimasi menggunakan *evolution strategies* guna membentuk persamaan untuk menentukan kebutuhan volume air berdasarkan kelembapan tanah.

1.4.2 Bagi Instansi

Laboratorium Benih BPTP Jatim memiliki seperangkat alat yang dapat memprediksi kebutuhan volume air berdasarkan kelembapan tanah secara otomatis untuk menjaga kelembapan media penumbuhan benih.

1.5 Batasan masalah

Agar pembahasan penelitian ini tidak menyimpang dari apa yang telah dirumuskan dan hasilnya dapat digunakan pada alat yang sedang dikembangkan di Laboratorium Benih BPTP Jatim, maka diperlukan batasan masalah. Batasan masalah dalam penelitian ini adalah:

1. Metode yang dipakai adalah regresi linier yang dioptimasi menggunakan *evolution strategies* guna membentuk persamaan regresi yang lebih kuat untuk melakukan prediksi.
2. Kelembapan tanah ideal sebagai acuan penelitian adalah pengetahuan pakar tentang kelembapan tanah ideal untuk menumbuhkan benih.
3. Data yang digunakan pada penelitian ini adalah data kelembapan tanah dan kebutuhan volume air yang diperoleh dari Laboratorium Benih BPTP Jatim.

1.6 Sistematika pembahasan

Untuk mencapai tujuan yang diharapkan, maka sistematika penulisan yang disusun dalam penelitian ini adalah sebagai berikut:

1.6.1 BAB I Pendahuluan

Pada bab ini akan dijelaskan tentang permasalahan yang menjadi latar belakang penelitian. Pada bagian latar belakang selain membahas tentang permasalahan juga akan mengulas sedikit tentang metode yang digunakan. Bab ini juga akan membahas tentang masalah yang sudah dirumuskan dari latar belakang serta tujuan, manfaat, batasan dan sistematika pembahasan.

1.6.2 BAB II Landasan Pustaka

Pada bab ini akan dibahas tentang kajian pustaka serta metode yang digunakan dalam penelitian ini. Bab ini juga akan menerangkan teknis yang digunakan untuk implementasi metode.

1.6.3 BAB III Metodologi

Pada bab ini akan dijelaskan tahapan metodologi yang dilakukan peneliti dalam melakukan rangkaian penelitian. Rangkaian penelitian tersebut dimulai dari studi pustaka hingga penarikan kesimpulan dan saran berdasarkan hasil penelitian.

1.6.4 BAB IV Perancangan

Bab ini menjelaskan tentang perancangan alur kerja sistem yang akan dibuat guna membuktikan kebenaran studi pustaka pada objek yang diangkat. Perancangan pada penelitian ini dibuat sebagai pedoman dalam implementasi sistem, agar sistem yang nantinya dibuat tidak menyimpang dari pembahasan. Pada bab ini akan dijelaskan lebih detail mengenai deskripsi umum sistem, batasan sistem, alur kerja sistem, perhitungan manual, perancangan tampilan antarmuka dan perancangan pengujian.

1.6.5 BAB V Implementasi

Bab ini menjelaskan tentang implementasi sistem berdasarkan metode yang digunakan pada penelitian ini. Implementasi pada penelitian ini bertujuan untuk membuat sistem yang lebih dinamis sehingga dapat mengolah data sesuai dengan metode yang digunakan. Pada bab ini akan dibahas secara detail tentang sistem yang dibuat beserta tampilan antarmuka dan perangkat yang digunakan.

1.6.6 BAB VI Pengujian dan Analisis

Bab ini menjelaskan hasil pengujian dan analisis terhadap implementasi sistem untuk membuktikan kebenaran metode dari kajian pustaka yang dibahas pada bab sebelumnya. Pada penelitian ini pengujian akan dibagi menjadi 2 bagian yaitu pengujian parameter *evolution strategies* serta pengujian kekuatan persamaan regresi dengan dan tanpa optimasi. Pada setiap bagian, analisis akan dilakukan untuk menarik kesimpulan hasil pengujian.

1.6.7 BAB VII Penutup

Bab ini merupakan bab terakhir yang membahas kesimpulan beserta saran untuk penelitian ini. Bab ini akan menjelaskan hasil dari penelitian yang sudah disimpulkan berdasarkan rumusan masalah. Selain hasil penelitian, bab ini juga akan membahas tentang saran untuk memperbaiki kekurangan penelitian ini guna menjadi masukan pada penelitian berikutnya.

BAB 2 LANDASAN PUSTAKA

Pada bab ini akan dibahas tentang kajian pustaka serta metode yang digunakan dalam penelitian ini. Bab ini juga akan menerangkan teknis yang digunakan untuk implementasi metode.

2.1 Kajian Pustaka

Pada bagian ini akan dibahas tentang beberapa pustaka terkait metode yang relevan dengan penelitian ini. Hal ini dilakukan agar beberapa pustaka tersebut dapat dijadikan landasan teori serta pembahasan tidak menyimpang dari rumusan masalah yaitu menerapkan metode yang digunakan terhadap objek penelitian. Dari pemaparan tersebut maka kajian pustaka ini akan membahas tentang regresi linier serta cara untuk mengoptimalkan hasil yang didapatkan dari metode tersebut. Berikut beberapa pustaka tentang metode relevan yang diperoleh dari berbagai sumber antara lain jurnal, buku dan penelitian sebelumnya.

Penelitian pertama membahas tentang regresi linier dan cara melakukan prediksi menggunakan metode tersebut. Pada penelitian tersebut regresi linier dipaparkan sebagai metode statistik yang cocok digunakan untuk menghitung pola hubungan antara variabel terikat dengan satu atau lebih variabel bebas. Pola ini dihitung untuk dijadikan sebuah persamaan matematik yang dikenal dengan persamaan regresi. Cara melakukan prediksi dengan regresi linier yaitu menggunakan persamaan tersebut untuk menghitung nilai variabel terikat prediksi berdasarkan variabel bebas yang dimasukkan (Levin & Rubin, 1998).

Penelitian kedua membahas tentang amatan pencilan (*outlier*) pada regresi linier. Keberadaan amatan pencilan pada data dapat menyebabkan persamaan regresi yang terbentuk menjadi salah. Hal ini dapat menyebabkan nilai sisaan (*error*) untuk masing-masing amatan menjadi besar sehingga persamaan regresi kurang kuat untuk mewakili pola hubungan variabel bebas dan variabel terikat. Kekuatan persamaan regresi yang terbentuk dapat dinilai dengan Kriteria Informasi. Nilai Kriteria Informasi yang semakin besar menandakan lemahnya persamaan regresi yang terbentuk dikarenakan adanya amatan pencilan di dalam data. Dari pemaparan tersebut, sebelum membentuk persamaan regresi sebaiknya amatan pencilan dihilangkan terlebih dahulu untuk memperkuat persamaan regresi yang dihasilkan (Johnson & Wichern, 1996).

Penelitian ketiga membahas tentang cara menghilangkan amatan pencilan pada regresi linier. Amatan pencilan tidak selalu bisa dideteksi secara langsung hanya dengan melihat grafik hubungan variabel bebas dan variabel terikat. Hal ini dikarenakan dalam data yang terlihat memiliki pola garis lurus bisa saja terdapat amatan pencilan, tergantung dari nilai variabel amatan lainnya. Dalam statistika, amatan berpengaruh sebagai pencilan jika memiliki nilai sisaan baku (*standardized error*) lebih dari 1.96 atau kurang dari -1.96 (Draper & Smith, 1998).

Penelitian keempat membahas prosedur dalam menghilangkan amatan pencilan dalam regresi linier. Mendeteksi amatan pencilan dengan sisaan baku bisa jadi melalui beberapa proses yang berulang. Hal ini disebabkan setelah amatan yang dinilai pencilan oleh sisaan baku dihilangkan, persamaan regresi akan berubah. Hal ini menyebabkan amatan yang sebelumnya bukan pencilan dapat menjadi pencilan baru. Dari pemaparan tersebut dapat ditarik kesimpulan bahwa pembentukan persamaan regresi serta perhitungan nilai sisaan baku masing-masing amatan harus dilakukan kembali setelah amatan yang dinilai pencilan pada proses sebelumnya dihilangkan. Proses tersebut harus terus dilakukan hingga amatan pencilan tidak terdeteksi lagi (Hadi & Simonoff, 1993).

Penelitian kelima membahas metode yang lebih efisien untuk mendeteksi amatan pencilan dalam regresi linier. Metode ini dikembangkan dari rumus *bayesian information criterion* (Schwarz, 1978) untuk disesuaikan dengan regresi linier. Hasil pengujian penelitian tersebut terhadap beberapa data menunjukkan bahwa kriteria informasi akan memberikan nilai yang baik apabila amatan berpotensi sebagai pencilan dihilangkan serta menyisakan amatan yang bukan pencilan sebanyak mungkin. Kesimpulan dari penelitian tersebut kriteria informasi yang telah disesuaikan untuk regresi linier dapat mendeteksi pencilan tanpa perlu menghitung sisaan baku, namun dibutuhkan metode optimasi untuk mencari nilai kriteria informasi terbaik (Tolvi, 2004).

Penelitian terakhir menerapkan metode optimasi *evolution strategies* untuk mencari kombinasi yang memiliki nilai kriteria informasi terbaik dengan cepat. Pada penelitian tersebut dipaparkan bahwa jumlah kemungkinan yang harus dicoba untuk mencari nilai kriteria informasi sebanyak $2^n - 1$ dimana n merupakan jumlah amatan pada data. Dari pemaparan tersebut penelitian tersebut menggunakan *evolution strategies* untuk mencari solusi terbaik dengan nilai kriteria informasi sebagai *fitness* sehingga tidak perlu mencoba seluruh solusi. Kesimpulan dari penelitian tersebut bahwa *evolution strategies* lebih efisien untuk digunakan dalam mendeteksi amatan pencilan, dikarenakan prosesnya dilalui tanpa perlu menghitung sisaan baku (Alma, Kurt, & Ugur, 2008).

2.2 Regresi Linier

Regresi linier adalah metode statistik yang cocok digunakan untuk menghitung pola hubungan variabel terikat dengan satu atau lebih variabel bebas pada masing-masing amatan. Regresi linier mampu melakukan pendekatan pola dalam bentuk persamaan melalui terbentuknya suatu hubungan yang bersifat numerik. Regresi linier juga dapat digunakan untuk melakukan prediksi pada variabel terikat, dengan cara memasukkan variabel bebas yang ingin dicari nilai variabel terikatnya ke dalam persamaan tersebut. Variabel bebas adalah variabel pengaruh, sedangkan variabel terikat merupakan variabel yang dipengaruhi (Levin & Rubin, 1998).

Namun yang perlu diingat, saat melakukan prediksi menggunakan persamaan regresi variabel bebas yang dimasukkan harus berada pada rentang data yang digunakan. Misalnya, persamaan regresi diperoleh menggunakan data dengan variabel bebas 1 sampai 100, maka variabel bebas yang kita gunakan untuk melakukan prediksi hanya boleh antara 1 sampai 100. Konsep ini disebut sebagai interpolasi (R Development Core Team, 2008).

Untuk melakukan prediksi menggunakan regresi linier, terlebih dahulu harus dicari persamaan regresinya (Kenney & Keeping, 1962). Rumus persamaan regresi linier untuk satu variabel bebas ditunjukkan pada persamaan berikut:

$$\hat{Y} = a + bX \quad (2.1)$$

$$a = \frac{(\sum Y)(\sum X^2) - (\sum X)(\sum XY)}{n(\sum X^2) - (\sum X)^2} \quad (2.2)$$

$$b = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum X^2) - (\sum X)^2} \quad (2.3)$$

Dimana \hat{Y} =variabel terikat hasil prediksi yang didapatkan dengan memasukkan variabel bebas (X) pada persamaan regresi, X =variabel bebas, Y =variabel terikat, a =konstanta, b =koefisien regresi (kemiringan) dan n =jumlah amatan.

2.3 Optimasi Regresi Linier

Dalam regresi linier, bisa jadi persamaan yang terbentuk kurang kuat untuk menggambarkan pola hubungan antar amatan. Persamaan regresi yang kurang kuat dapat menyebabkan prediksi menjadi salah. Hal ini dikarenakan di dalam data terdapat beberapa amatan pencilan yang ikut membentuk persamaan regresi. Amatan pencilan (*outlier*) merupakan amatan yang memiliki nilai terlalu jauh dari amatan lainnya. Dari pemaparan di atas, diperlukan optimasi untuk memperkuat persamaan regresi dengan cara mendeteksi amatan pencilan guna dihilangkan sebelum membentuk persamaan tersebut (Johnson & Wichern, 1996).

Cara yang biasa dilakukan untuk mendeteksi amatan pencilan yaitu membandingkan sisaan pada setiap amatan, namun cara ini kurang efisien karena bisa jadi membutuhkan proses berulang. Pada penelitian ini amatan pencilan dideteksi menggunakan *evolution strategies* dengan kriteria informasi sebagai *fitness*, dikarenakan cara ini lebih efisien daripada membandingkan nilai sisaan baku (Alma, Kurt, & Ugur, 2008).

2.3.1 Sisaan (Error)

Amatan pencilan tidak selalu bisa dideteksi secara langsung hanya dengan melihat grafik hubungan variabel bebas dan variabel terikat. Hal ini dikarenakan dalam data yang terlihat memiliki pola garis lurus bisa saja terdapat amatan pencilan, tergantung dari nilai variabel amatan lainnya. Cara yang paling sering dilakukan untuk mendeteksi amatan pencilan adalah dengan membandingkan

nilai sisaan dari masing-masing amatan. Sisaan (*error*) merupakan selisih antara variabel terikat pada data amatan ke-*i* dengan prediksi variabel terikat menggunakan variabel bebas pada amatan yang sama. Semakin besar nilai sisaan menunjukkan bahwa penyimpangan amatan ke-*i* juga semakin besar. Dalam statistika, amatan dikatakan pencilan jika memiliki nilai sisaan baku (*standardized error*) lebih dari 1.96 atau kurang dari -1.96 (Draper & Smith, 1998).

Dalam suatu data bisa jadi terdapat lebih dari satu amatan pencilan. Ketika amatan pencilan dihilangkan akan terbentuk persamaan regresi baru, hal ini dapat menyebabkan amatan yang tadinya bukan pencilan menjadi pencilan baru. Pendeteksian pencilan dengan metode ini bisa jadi membutuhkan proses berulang, yang mana satu proses membutuhkan beberapa tahap antara lain mencari persamaan regresi, sisaan dan sisaan baku (Hadi & Simonoff, 1993). Hal ini menyebabkan pendeteksian pencilan menjadi kurang efisien, maka perlu sebuah metode yang dapat mendeteksi pencilan secara bersamaan.

2.3.2 Kriteria Informasi

Kriteria informasi (*KI*) dapat mendeteksi amatan pencilan secara bersamaan dengan cara menghitung kualitas suatu data yang dihasilkan setelah satu atau beberapa amatan tidak disertakan dalam perhitungan. Nilai kriteria informasi akan semakin kecil jika semua amatan berpotensi sebagai pencilan tidak disertakan dan menyertakan amatan yang tergolong bukan pencilan sebanyak mungkin, artinya nilai kriteria informasi terbaik adalah nilai kriteria informasi paling kecil. Kriteria informasi juga dapat digunakan untuk mendeteksi amatan pencilan pada regresi linier dengan menggunakan konsep sisaan (Mills & Prasad, 1992). Rumus kriteria informasi ditunjukkan pada Persamaan 2.4.

$$KI = \log \hat{\sigma}^2 + (1 + p + m_d) \frac{\log n}{n} \quad (2.4)$$

Dimana *KI*=nilai kriteria informasi, $\hat{\sigma}^2$ =penduga ragam model regresi dari semua amatan yang dianggap bukan pencilan, *p*=jumlah variabel bebas, *m_d*=jumlah amatan yang dianggap pencilan dan *n*=jumlah amatan.

Nilai kriteria informasi di atas mampu menentukan amatan mana saja yang merupakan pencilan guna menghasilkan persamaan regresi terkuat yang mungkin dicapai. Hal ini merupakan kelemahan yang membuat amatan bukan pencilan terlalu sedikit. Kelemahan ini harus diatasi guna memperoleh keseimbangan antara mencari memperkuat persamaan regresi namun menyisakan amatan yang bukan pencilan sebanyak mungkin. Dari pemaparan tersebut maka rumus kriteria informasi ini diperbaiki dengan menambahkan nilai penalti sebagai titik berat dalam mendeteksi amatan pencilan. Nilai penalti yang semakin kecil akan menitikberatkan seleksi untuk memperkuat persamaan regresi, namun nilai penalti terkecil dibatasi lebih dari 1. Sebaliknya semakin besar nilai penalti akan menitikberatkan banyaknya jumlah amatan dianggap bukan pencilan dalam

seleksi (Tolvi, 2004). Dalam penelitian ini nilai penalti yang digunakan adalah 1.5 dengan hasil seleksi yang dijabarkan pada subbab pengujian. Hal ini dilakukan untuk membuat pendeteksian amatan pencilan menggunakan kriteria informasi memiliki hasil yang sama dengan menggunakan sisaan baku. Rumus kriteria informasi yang telah diperbaiki dengan menambahkan nilai penalti ditunjukkan pada Persamaan 2.5.

$$KI = \log \hat{\sigma}^2 + (1 + p) \frac{\log n}{n} + Km_d \frac{\log n}{n} \quad (2.5)$$

Dimana KI =nilai kriteria informasi, $\hat{\sigma}^2$ =penduga ragam model regresi dari semua amatan yang dianggap bukan pencilan, K =nilai penalti, p =jumlah variabel bebas, m_d =jumlah amatan yang dianggap pencilan dan n =jumlah amatan.

Kriteria informasi untuk regresi linier membutuhkan penduga ragam model regresi. Penduga ragam model regresi ini diperoleh dari sisaan menggunakan persamaan regresi dengan menyertakan seluruh amatan, namun hanya dijumlahkan yang dianggap bukan pencilan saja. Hal ini dilakukan untuk menghindari hasil perhitungan penduga ragam model regresi memiliki nilai 0, yang berakibat kriteria informasi tidak dapat menyisakan amatan bukan pencilan sebanyak mungkin. Hal ini terjadi jika pola data lurus sempurna apabila amatan pencilan dihilangkan. Rumus untuk menghitung penduga ragam model regresi guna mencari nilai kriteria informasi ditunjukkan pada Persamaan 2.6.

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n-m_d} (e_i^2)}{n - m_d} \quad (2.6)$$

Dimana $\hat{\sigma}^2$ = penduga ragam model regresi dari semua amatan yang dianggap bukan pencilan, n =jumlah amatan bukan pencilan, m_d =jumlah amatan pencilan dan e_i =sisaan amatan ke- i .

Untuk mendapatkan nilai kriteria informasi terbaik, semua kemungkinan kombinasi amatan mana saja yang dianggap pencilan dan bukan harus dicoba. Pada penelitian ini kemungkinan tersebut selanjutnya akan disebut sebagai solusi. Agar penggunaan kriteria informasi ini lebih efisien maka diperlukan metode yang dapat mencari solusi terbaik dengan cepat menggunakan nilai kriteria informasi sebagai pembanding (Alma, Kurt, & Ugur, 2008).

2.4 Evolution strategies

Algoritma *Evolution strategies* mulai dikenal pada tahun 1960-an dimana algoritma ES paling sering digunakan sebagai solusi dari permasalahan *black-box* pada ruang pencarian terus menerus (Hansen, Arnold, & Auger, 2014). Algoritma *Evolution strategies* merupakan bagian dari algoritma evolusi yang menerapkan proses mutasi, rekombinasi, dan seleksi pada populasi calon solusi dari permasalahan yang ada.

Algoritma ES lebih mengutamakan representasi bilangan pecahan (*real-vector*) pada representasi kromosomnya. Algoritma ES memiliki siklus atau metode yang mirip dengan algoritma genetika. Perbedaan antara algoritma ES dan algoritma genetika terletak pada proses reproduksinya. Algoritma genetika lebih mengutamakan proses *mutation* sebagai operator reproduksinya dan proses reproduksi sebagai penunjangnya. Algoritma ES lebih mengutamakan proses mutasi sebagai operator reproduksinya sedangkan proses rekombinasi dijadikan sebagai opsi tambahan. Pada siklus algoritma ES dikenal juga mekanisme *self-adaption*. Mekanisme *self-adaption* ini digunakan sebagai kontrol parameter yang mengatur perubahan nilai parameter pencariannya (Mahmudy, 2015).

2.4.1 Struktur Dasar Algoritma *Evolution strategies*

Algoritma ES memiliki beberapa notasi dan istilah yang akan digunakan dalam siklus algoritma ES. Berikut ini akan dijelaskan notasi yang sering digunakan dalam algoritma ES:

1. μ (*miu*), menyatakan banyaknya populasi induk yang terlibat selama siklus ES.
2. λ (*lambda*), menyatakan banyaknya *offspring* yang akan dihasilkan oleh induk selama proses reproduksi.
3. R (rekombinasi), menyatakan banyaknya induk yang akan diikutsertakan pada proses rekombinasi untuk menghasilkan satu *offspring*.
4. σ (*sigma*), merupakan bilangan acak yang dibangkitkan pada rentang $[0,1]$.
5. $N(0,1)$, merupakan bilangan acak yang didapat menggunakan rumus tertentu, dimana bilangan acak tersebut memiliki rata-rata sebesar 0 dan standar deviasi sebesar 1.

Algoritma ES lebih mengutamakan operator mutasi pada proses reproduksinya sedangkan rekombinasi dijadikan sebagai opsi tambahan, yang dapat digunakan ataupun tidak digunakan pada proses reproduksi. Karena hal tersebut, algoritma ES memiliki empat tipe siklus yang akan dijelaskan pada Tabel 2.1.

Tabel 2. 1 Siklus Algoritma ES

No.	Tipe Siklus Algoritma ES	Deskripsi
1	ES (μ, λ)	Siklus ES (μ, λ) tidak menggunakan Operator rekombinasi pada tahapan reproduksinya. Tahapan seleksi, siklus ES (μ, λ) hanya melibatkan <i>offspring</i>

		sebagai kandidat individu yang akan diseleksi menjadi individu baru pada generasi berikutnya.
2	ES ($\mu/R, \lambda$)	Siklus ES ($\mu/R, \lambda$) menggunakan operator rekombinasi pada tahapan reproduksinya. Pada tahapan seleksi siklus ES ($\mu/R, \lambda$) hanya melibatkan <i>offspring</i> sebagai kandidat individu yang akan diseleksi menjadi individu baru pada generasi berikutnya.
3	ES ($\mu + \lambda$)	Siklus ES ($\mu + \lambda$) tidak menggunakan operator rekombinasi pada tahapan reproduksinya. Pada tahapan seleksi, siklus ES ($\mu + \lambda$) melibatkan <i>offspring</i> dan induk sebagai kandidat individu yang akan diseleksi menjadi individu baru pada generasi berikutnya.
4	ES ($\mu/R + \lambda$)	Siklus ES ($\mu/R + \lambda$) menggunakan operator rekombinasi pada tahapan reproduksinya. Pada tahapan seleksi, siklus ES ($\mu/R + \lambda$) melibatkan <i>offspring</i> dan induk sebagai kandidat individu yang akan diseleksi menjadi individu baru pada generasi berikutnya.

Sumber: dasar-dasar algoritma evolusi (wayan, 2016)

Algoritma ES memiliki siklus yang mirip dengan algoritma evolusi lainnya, yakni algoritma ES juga menerapkan proses inisialisasi, reproduksi, evaluasi, dan seleksi. Siklus algoritma ES akan dideskripsikan melalui pseudocode pada Tabel 2.2.

Tabel 2. 2 Psudocode ES ($\mu/r +, \lambda$)

Baris	Prosedur ES ($\mu/r +, \lambda$)
1.	Begin
2.	$t := 0$
3.	inisialisasi $P(t) := \{(p_m(0), q_m(0)), m=1, \dots, \mu\}$;
4.	while (bukan kondisi berhenti) do
5.	for $i=1$ to λ do begin
6.	$\alpha_i := \text{pilihInduk}(P(t), r)$;
7.	$p_i := \text{rekombinasi}(\alpha_i)$;
8.	$q_i := \text{rekombinasi}(\alpha_i)$;
9.	$p'_i := \text{mutasi}(p_i)$;
10.	$q'_i := \text{mutasi}(q_i)$;
11.	$F_i := \text{anak}(p'_i, q'_i)$;
12.	end
13.	$C(t) := \{(p_l(t), q_l(t)), l=1, \dots, \lambda\}$;
14.	case tipe_seleksi of

15.	case ($\mu/r, \lambda$):
16.	$P(t+1) := \text{seleksi}(C(t), \mu);$
17.	case ($\mu/r + \lambda$):
18.	$P(t+1) := \text{seleksi}(P(t), C(t), \mu);$
19.	end
20.	$t := t + 1;$
21.	end while
22.	End

Pada tahap awal siklus ES dideklarasikan terlebih dahulu urutan generasinya, pada pseudocode urutan generasi dilambangkan dengan variabel t . Setelah itu, populasi awal diinisialisasi sebanyak μ yang telah ditentukan. Populasi awal ini disebut sebagai parent (induk) yang dilambangkan dengan $P(t)$ pada pseudocode. Notasi $P(t)$ dibaca sebagai "induk pada generasi ke- t ". Variabel p_m dan q_m merupakan parameter objek yang digunakan pada proses reproduksi. Langkah selanjutnya dilakukan perulangan sampai memenuhi kondisi untuk berhenti (baris 4-21). Pada perulangan tersebut, terdapat perulangan baru untuk melakukan proses reproduksi (baris 5-12). Perulangan dilakukan sebanyak λ yang telah ditentukan. Dalam proses reproduksi, langkah pertama yang dilakukan ialah memilih induk sebanyak r yang telah ditentukan dari μ induk yang ada. Sejumlah induk yang terpilih tersebut akan dilakukan proses rekombinasi berdasarkan banyak parameter objeknya (baris 7-8). Individu yang dihasilkan selama proses rekombinasi akan dijadikan sebagai induk pada proses mutasi (baris 9-10). Setelah itu, individu dari hasil mutasi akan dijadikan sebagai individu anak. Setelah proses reproduksi dilakukan, maka individu anak yang dihasilkan pada proses reproduksi akan disimpan pada variabel $C(t)$. Tahap berikutnya adalah melakukan seleksi. Proses seleksi akan dilakukan berdasarkan tipe siklusnya. Jika siklus yang digunakan adalah ES ($\mu/r, \lambda$) maka induk tidak terlibat dalam proses seleksi. Jika siklus yang digunakan adalah ES ($\mu/r + \lambda$) maka induk ikut terlibat dalam proses seleksi. Setelah proses seleksi dilakukan maka didapatkan populasi induk yang baru $P(t+1)$.

Siklus ES merupakan sebuah proses yang akan dilakukan terus menerus sampai memenuhi kondisi untuk berhenti. Pada umumnya, kondisi standar yang sering digunakan untuk memberhentikan siklus ES ialah (Hansen, Arnold, & Auger, 2014):

- a. Persyaratan sumber daya
 1. Siklus ES sudah mencapai maksimum generasi yang telah ditentukan.
 2. Siklus ES sudah mencapai maksimum waktu CPU bekerja (CPU-time).
- b. Persyaratan konvergensi
 1. Nilai *fitness* mengalami konvergensi, yakni nilai *fitness* yang dihasilkan tidak mengalami perubahan yang signifikan atau tidak mengalami perubahan sama sekali.
 2. Nilai pada parameter objek mengalami konvergensi, yakni nilai parameter objek yang dihasilkan tidak mengalami perubahan yang signifikan atau tidak mengalami perubahan sama sekali.

3. Nilai pada parameter strategi mengalami konvergensi, yakni nilai parameter strategi yang dihasilkan tidak mengalami perubahan yang signifikan atau tidak mengalami perubahan sama sekali.

2.4.2 Representasi Kromosom

Representasi kromosom merupakan representasi dari solusi yang akan dihasilkan dalam siklus ES. Satu buah kromosom tersusun atas berbagai gen yang merepresentasikan masing-masing solusi. Seperti yang telah dikatakan sebelumnya bahwa siklus algoritma ES lebih mengutamakan representasi bilangan pecahan (*real-vector*) sebagai representasi kromosomnya. Namun, seiring perkembangan yang berjalan, algoritma ES juga dapat menggunakan representasi permutasi untuk memecahkan permasalahan kombinatorial

a. Representasi Bilangan Pecahan(Real-Vector)

Pada representasi *real-vector* terdapat parameter tambahan selain dari pada parameter objeknya. Parameter tambahan ini dilambangkan dengan σ (*sigma*) yang melekat pada setiap gen (parameter objek) yang ada. Mekanisme *self-adaption* pada siklus algoritma ES akan merubah nilai σ (*sigma*) secara adaptif di setiap pertambahan generasi. Jika P dilambangkan dengan individu induk maka P mempunyai susunan kromosom pada representasi *real-vector* seperti yang ditunjukkan pada Gambar 2.1

$$P = (x_1, x_2, \sigma_1, \sigma_2)$$

$$P = (2.0123, 3.1245, 0.324, 0.879)$$

Gambar 2. 1 Representasi Real-Vector

Gambar 2.1 menunjukkan bahwa kromosom P memiliki dua buah parameter objek yakni x_1 dan x_2 yang memiliki nilai σ (*sigma*) masing – masing satu buah (σ_1 dan σ_2). Nilai x_1 dan x_2 dibangkitkan secara acak berdasarkan rentang nilai tertentu sedangkan nilai σ_1 dan σ_2 dibangkitkan secara acak dengan rentang nilai [0,1].

b. Representasi Permutasi

Representasi permutasi merupakan representasi kromosom yang melambangkan setiap gennya dengan bilangan *integer*. Jika sebuah kromosom tersusun atas lima buah gen, maka setiap gennya akan dilambangkan dengan bilangan *integer* 1 sampai dengan 5. Oleh karena itu, pada representasi permutasi

tidak diperkenankan memiliki nilai ganda di dalam kromosomnya. Mekanisme *self-adaption* sering tidak dipakai jika menggunakan representasi permutasi. Namun, jika ingin tetap mengadopsi mekanisme *self-adaption*, dapat dipakai pada representasi permutasi yang menggunakan metode *reciprocal exchange mutation* atau *insertion mutation* (Mahmudy, 2015). Bentuk representasi permutasi pada siklus algoritma ES akan ditunjukkan pada Gambar 2.2

P(t)	Representasi Permutasi	σ (sigma)
P ₁	[3, 1, 4, 2, 5]	1.45789

Gambar 2. 2 Representasi Permutasi

Gambar 2.2 menunjukkan bahwa kromosom P₁ memiliki lima buah parameter objek (gen) yang dilambangkan dengan angka 1 sampai dengan 5. Kelima gen tersebut disusun secara acak untuk membentuk sebuah kromosom yang memiliki panjang string sebesar 5. Nilai σ (sigma) dibangkitkan secara acak pada rentang nilai tertentu. Pada kasus yang ditunjukkan oleh Gambar 2.3, nilai σ (sigma) menyatakan jumlah proses *exchange mutation* atau *insertion mutation* yang akan dilakukan untuk menghasilkan satu anak.

2.4.3 Inisialisasi

Tahap inisialisasi merupakan tahap pembangkitan individu pada generasi pertama secara acak. Pada proses inisialisasi akan ditentukan jumlah populasi awal (μ) dan banyaknya anak yang akan dihasilkan pada proses reproduksi (λ). Setelah itu, kromosom-kromosom akan dibangkitkan sebanyak μ yang telah ditentukan. Dalam kromosom tersebut, parameter-parameter objek akan dibangkitkan berdasarkan rentang nilai tertentu begitu juga dengan nilai sigma yang melekat pada setiap parameter objek. Pada tahap ini, nilai fitness dari setiap individu juga dihitung berdasarkan rumusan yang telah ditetapkan.

2.4.4 Representasi

Tahap reproduksi merupakan tahapan yang dilakukan untuk menghasilkan individu-individu yang baru. Pada siklus algoritma ES, tahap reproduksi lebih mengutamakan proses mutasi sedangkan rekombinasi hanya dijadikan sebagai opsi tambahan yang dapat digunakan ataupun tidak. Namun, tidak berarti proses rekombinasi dapat diabaikan begitu saja. Individu baru yang didapatkan dari hasil reproduksi disebut sebagai *offspring*. Sebelum melakukan tahapan reproduksi, hal yang perlu diperhatikan ialah menentukan banyaknya individu anak (*offspring*) yang akan dihasilkan selama proses reproduksi. Banyaknya individu anak yang akan dihasilkan dapat ditentukan dengan rumus:

$$\lambda = C \times \mu \quad (2.7)$$

Keterangan:

λ (lamda) : Jumlah individu anak (*offspring*) yang akan dihasilkan pada reproduksi

C : Konstanta

μ (miu) : Jumlah populasi induk

a. Mutasi

Proses mutasi pada siklus algoritma ES mirip dengan proses mutasi pada algoritma genetika karena dalam proses mutasi hanya melibatkan satu individu induk untuk menghasilkan *offspring*. Representasi real-vector pada umumnya digunakan dalam siklus algoritma ES. Pada Representasi real-vector, setiap gen (parameter objek) yang membentuk kromosom mengalami proses mutasi.

Pada Gambar 2.2, setiap individu terdiri dari dua parameter objek yakni x_1 dan x_2 yang masing-masingnya memiliki nilai sigma yakni σ_1 dan σ_2 . Maka dari itu, untuk menghasilkan nilai setiap parameter objek pada *offspring* yang akan dihasilkan dapat menggunakan rumus:

$$x'_1 = x_1 + \sigma_1 N(0,1) \quad (2.8)$$

$$x'_2 = x_2 + \sigma_2 N(0,1) \quad (2.9)$$

$N(0,1)$ merupakan bilangan acak yang memiliki rata-rata sebesar 0 dan standar deviasi sebesar 1. $N(0,1)$ didapatkan dengan membangkitkan dua bilangan acak yakni r_1 dan r_2 kemudian menghitungnya dengan menggunakan rumus:

$$N(0,1) = \sqrt{-2 \ln r_1} \sin 2\pi r_2 \quad (2.10)$$

Karena siklus algoritma ES memiliki mekanisme self-adaption maka nilai sigma akan mengalami perubahan secara adaptif. Jika 20% individu baru yang dihasilkan oleh sebuah individu induk memiliki nilai fitness yang lebih tinggi dari nilai fitness induknya, maka nilai sigma akan meningkat 10% dari nilai awalnya. Hal yang sama akan terjadi, jika kurang dari 20% individu baru yang dihasilkan oleh sebuah individu induk memiliki nilai fitness yang lebih tinggi dari nilai fitness induknya, maka nilai sigma akan menurun 10% dari nilai awalnya.

2.4.5 Evaluasi

Tahap evaluasi dilakukan setelah menemukan individu baru (*offspring*) dari hasil reproduksi. Tahap evaluasi merupakan proses menghitung nilai kebugaran *fitness* dari setiap individu yang ada. Nilai *fitness* merupakan tolak ukur seberapa baik sebuah individu dapat dijadikan sebagai solusi dari permasalahan yang ada. Dalam menentukan nilai *fitness*, terdapat beberapa rumus yang berbeda tergantung pada solusi yang diinginkan. Untuk solusi dari permasalahan maksimasi, nilai *fitness* harus berbanding lurus dengan nilai $f(x)$. Perhitungan nilai *fitness* dapat menggunakan rumus:

$$fitness = f(x) \quad (2.11)$$

Untuk solusi dari permasalahan minimalisasi nilai $f(x)$, nilai *fitness* harus berbanding terbalik dengan nilai $f(x)$. Perhitungan nilai *fitness* dapat menggunakan rumus:

$$fitness = C - f(x) \quad (2.12)$$

$$fitness = \frac{1}{f(x)} \quad (2.13)$$

$$fitness = \frac{1}{f(x)+c} \quad (2.14)$$

Pada perhitungan nilai *fitness* untuk solusi minimasi terdapat variabel C yang merupakan sebuah konstanta untuk menghindari nilai 0 dan nilai tak terhingga pada hasil perhitungan nilai *fitness*.

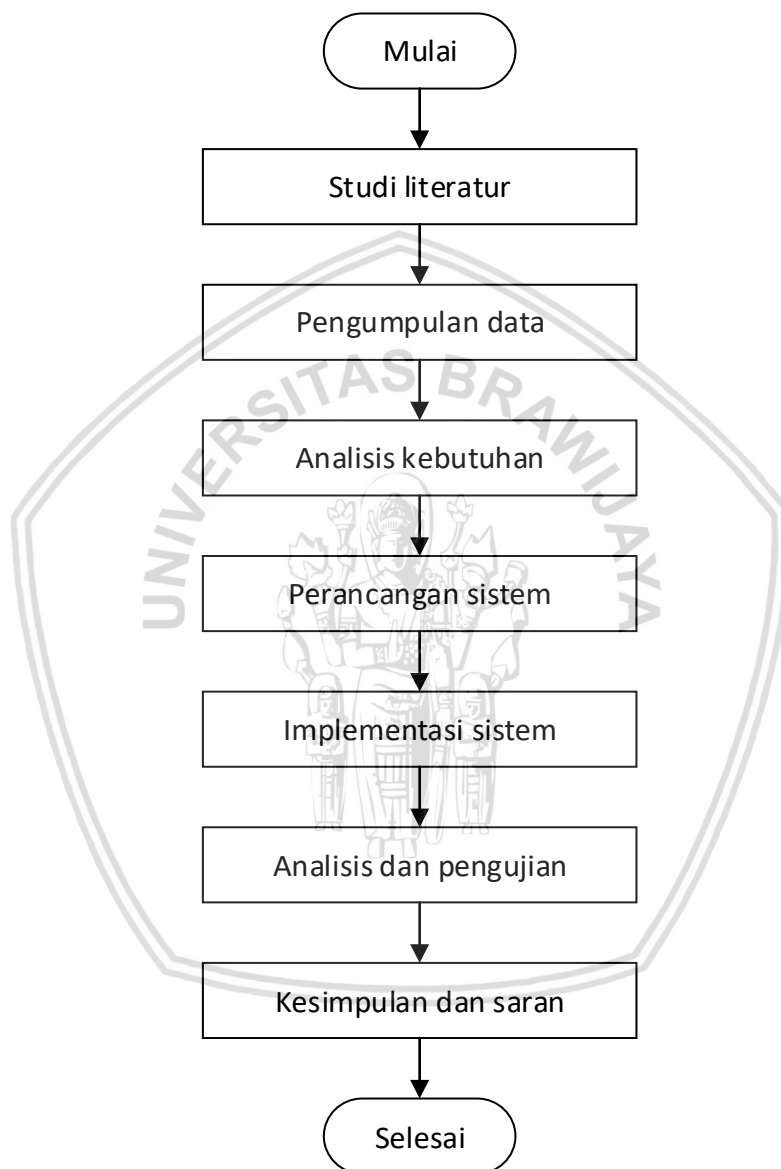
2.4.6 Seleksi

Berdasarkan tipe siklusnya, algoritma ES memiliki dua jenis seleksi yakni siklus (μ, λ) yang disebut comma selection dan siklus $(\mu + \lambda)$ yang disebut plus selection). Seleksi yang terjadi pada siklus comma selection tidak melibatkan individu induk sebagai kandidat individu yang akan diseleksi. Sebaliknya, seleksi yang terjadi pada siklus plus selection melibatkan individu induk dan *offspring* sebagai kandidat individu yang akan diseleksi untuk generasi berikutnya.

Dalam siklus algoritma ES, metode yang digunakan dalam seleksi ialah elitism selection. Seleksi dengan menggunakan metode elitism selection hanya memilih individu yang memiliki nilai *fitness* terbaik dari kandidat individu yang ada. Metode elitism selection menganggap individu dengan nilai *fitness* yang rendah merupakan individu yang buruk.

BAB 3 METODOLOGI

Pada bab ini akan dijelaskan tahapan metodologi yang dilakukan peneliti dalam melakukan rangkaian penelitian. Rangkaian penelitian tersebut dimulai dari studi pustaka hingga penarikan kesimpulan dan saran berdasarkan hasil penelitian. Tahapan metodologi penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tahap metodologi penelitian

Gambar 3.1 merupakan tahapan yang dilakukan oleh peneliti dalam mengimplementasikan *Evolution strategies* untuk Optimasi Pembentukan Fungsi Regresi Linier dalam Menentukan Kebutuhan Volume Air Penyiraman Tanah.

Tahapan tersebut dimulai dari studi literatur hingga penarikan kesimpulan dan saran berdasarkan hasil penelitian.

3.1 Studi Pustaka

Subbab ini menerangkan pustaka yang dipelajari atau dipakai dari beberapa konsentrasi ilmu pada berbagai sumber. Berikut pustaka yang digunakan dalam penelitian ini.

- a. Regresi linier sederhana dan cara melakukan prediksi dengan regresi linier.
- b. Amatan pencilan pada regresi linier dan cara menentukan kekuatan persamaan regresi.
- c. Kriteria informasi untuk mendeteksi amatan pencilan pada regresi linier.
- d. *Evolution strategies* menggunakan kriteria informasi sebagai *fitness* untuk menentukan amatan mana saja yang berpotensi sebagai pencilan.

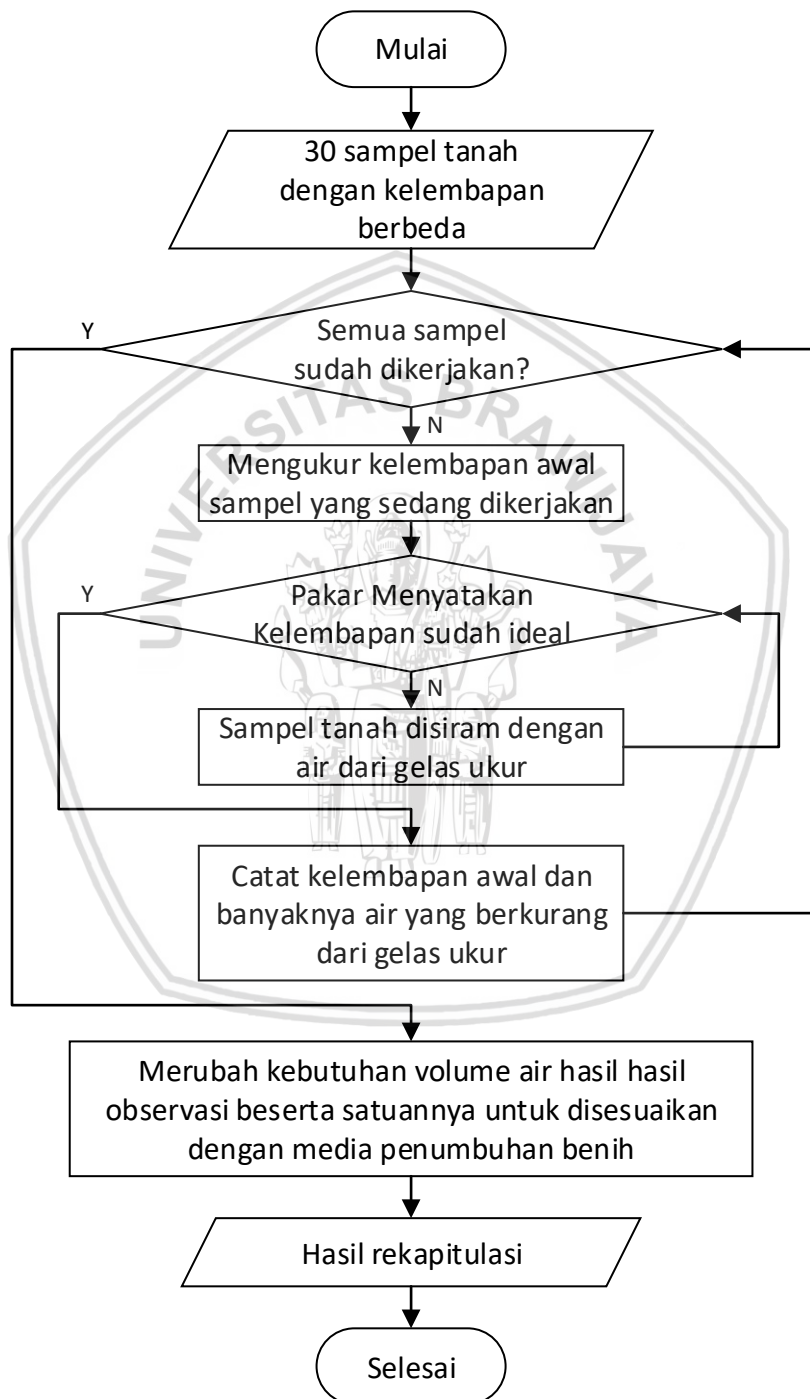
Pustaka di atas berasal dari berbagai sumber antara lain jurnal, buku dan penelitian sebelumnya dengan metode relevan. Pustaka ini merupakan dasar dari metode yang diangkat pada penelitian ini.

3.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data hubungan pengaruh kelembapan tanah awal terhadap kebutuhan volume air untuk mencapai kelembapan tanah ideal. Variabel bebas pada data ini adalah kelembapan tanah awal, sedangkan variabel terikatnya adalah volume air yang dibutuhkan. Data ini diperoleh dari observasi bersama pakar di Laboratorium Benih BPTP Jatim menggunakan alat siram otomatis yang sedang dikembangkan untuk dimanfaatkan sensornya guna mendapatkan kelembapan awal. Untuk meminimalkan ketidakakuratan hasil observasi maka observasi ini dilakukan terhadap 30 sampel tanah. Sampel tanah yang digunakan untuk observasi yaitu sampel tanah dalam keadaan kering hingga paling lembap pada kondisi sebelum penyiraman rutin. Sampel yang telah dikumpulkan tersebut kemudian diukur nilai kelembapan tanahnya menggunakan sensor kelembapan dalam satuan volt. Setelah itu masing-masing sampel tanah akan disiram air dalam gelas ukur oleh pakar hingga mencapai tingkat kelembapan ideal.

Setelah semua proses selesai dilakukan, kelembapan tanah awal dan kebutuhan air pada masing-masing sampel tanah akan direkapitulasi. Berdasarkan luasnya media penumbuhan benih yang akan mempersulit observasi untuk 30 sampel tanah, maka observasi dilakukan pada sampel tanah dengan volume 64 cm³ untuk masing-masing sampel. Menurut pakar hal ini dapat dilakukan karena perbandingan volume tanah dan kebutuhan air baik dengan volume tanah 64 cm³ maupun media penumbuhan benih yang memiliki volume tanah 125 kali lebih banyak yaitu 8000 cm³ akan sama. Dari pernyataan tersebut maka volume air yang dibutuhkan untuk membuat media penumbuhan benih mencapai kelembapan

ideal cukup mengalikan kebutuhan volume air hasil observasi 125 kali lebih banyak. Untuk mempermudah pembacaan data maka volume air yang dibutuhkan akan dirumah satuannya dari mililiter ke dalam satuan liter. Perubahan volume air dan satuannya untuk disesuaikan dengan media penumbuhan benih akan dilakukan sebelum rekapitulasi data. Diagram untuk menggambarkan proses yang dilalui guna mendapatkan hasil rekapitulasi data ditunjukkan pada Gambar 3.2.



Gambar 3.2 Proses observasi data

Gambar 3.2 merupakan proses dalam melakukan observasi guna memperoleh data. Pada proses di atas setelah sampel tanah diukur kelembapan awalnya maka pakar akan menyiram sampel tersebut dari gelas ukur. Volume air yang berkurang dari gelas ukur merupakan kebutuhan volume air untuk sampel tersebut. Berikut beberapa dokumentasi proses obeservasi guna memperoleh data yang ditunjukkan pada gambar 3.3 dan 3.4.



Gambar 3.3 Mengukur kelembapan awal sampel tanah

Gambar 3.3 merupakan dokumentasi proses dalam mengukur kelembapan awal sampel tanah. Sampel tanah diukur menggunakan sensor kelembapan dalam satuan volt. Berikut dokumentasi dalam menentukan volume air yang dibutuhkan untuk membuat kelembapan sampel tanah yang sudah diukur menjadi ideal.



Gambar 3.4 Pakar menuangkan air kepada sampel tanah

Gambar 3.4 merupakan dokumentasi proses ketika pakar menuangkan air dari gelas ukur pada sampel tanah untuk membuat sampel tersebut mencapai kelembapan ideal. Volume air yang berkurang dari gelas ukur akan menyatakan volume air yang dibutuhkan sampel tanah untuk mencapai kelembapan ideal.

3.3 Analisis Kebutuhan

Analisis kebutuhan pada penelitian ini dilakukan untuk menerangkan perangkat yang digunakan dalam perhitungan manual serta implementasi sistem. Daftar perangkat yang digunakan antara lain:

1. Perangkat keras
 - a. Laptop
 - Intel(R) Core(TM) i5-7200U CPU @ 2.20GHz (4CPUs)
 - RAM 4 GB
 - Harddisk kapasitas 1TB
 - Monitor 14 inch
 - b. Printer
2. Perangkat lunak
 - a. NetBeans
 - b. Sublime Text
 - c. Google Chrome
 - d. Microsoft Excel 2013
 - e. Microsoft Word 2013

Berikut perangkat yang digunakan untuk perhitungan manual dan implementasi sistem pada penelitian ini. Selain perhitungan manual serta

implementasi perangkat di atas juga digunakan oleh peneliti untuk melakukan analisis dan pengujian.

3.4 Perancangan Sistem

Subbab ini membahas rangkaian perancangan yang digunakan sebagai acuan langkah kerja pada proses implementasi. Perancangan yang digunakan antara lain mengenai deskripsi umum sistem, batasan sistem, alur kerja sistem, perhitungan manual, perancangan tampilan antarmuka dan perancangan pengujian.

3.4.1 Deskripsi Umum Sistem

Sistem yang dibangun mengimplementasikan *evolution strategies* untuk mengoptimasi regresi linier guna mencari persamaan regresi terbaik dengan jumlah amatan terbanyak pada data. Hasil keluaran sistem merupakan hasil perhitungan variable berdasarkan inputan yang dimasukan, yaitu hasil variansi, proses random ke-1, proses random ke-2, dan populasi hasil seleksi akhir. Kemudian selain menampilkan komponen tersebut sistem juga akan menampilkan seberapa kuat persamaan regresi yang dibentuk dalam ukuran nilai Kriteria Informasi. Di dalam penelitian ini juga digunakan metode optimasi, sehingga sistem juga akan memberikan beberapa pilihan komponen-komponen optimasi pada *evolution strategies* sebelum melakukan pemrosesan data seperti jumlah populasi, jumlah offspring dan jumlah generasi.

3.4.2 Arsitektur Sistem

Secara terstruktur sistem yang dibangun pada penelitian ini akan dibagi menjadi 3 bagian yaitu tampilan, proses dan penyimpanan. Pada bagian tampilan akan memuat hasil keluaran dan masukan yang dibutuhkan oleh sistem. Bagian proses akan mengolah masukan menggunakan rumus serta data yang digunakan untuk menghasilkan keluaran. Selanjutnya bagian penyimpanan akan mengatur penyimpanan data dan hasil proses perhitungan metode.

3.5 Implementasi Sistem

Pada bagian implementasi sistem akan dibahas bagaimana tahapan membangun suatu sistem yang mengacu pada perancangan sistem yang sudah dibuat sebelumnya. Tahapan yang ada dalam implementasi antara lain:

1. Implementasi tampilan menggunakan bahasa pemrograman java yang akan diolah menggunakan Netbeans.
2. Implementasi penyimpanan menggunakan *array* dan *integer* yang diolah dalam bahasa pemrograman java.
3. Implementasi perhitungan regresi linier dioptimasi menggunakan *evolution strategies* menggunakan bahasa pemrograman java yang akan dijalankan dengan program netbeans.

3.6 Pengujian dan Analisis

Bagian ini membahas tentang pengujian dan analisis yang dilakukan terhadap sistem yang dibuat pada penelitian ini. Pengujian pada penelitian ini akan dilakukan terhadap parameter *evolution strategies* serta akurasi persamaan regresi yang terbentuk dengan dan tanpa optimasi. Berdasarkan pemaparan tersebut maka pengujian dan analisis yang dilakukan pada penelitian ini adalah pengujian dan analisis parameter *evolution strategies* paling efektif dan efisien untuk mendapatkan *fitness* terbaik.

Hasil *fitness* terbaik yang tercatat dari pengujian parameter *evolution strategies* akan digunakan sebagai hasil optimasi pada pengujian kekuatan persamaan regresi.

3.7 Kesimpulan dan Saran

Pengambilan kesimpulan akan dilaksanakan ketika semua tahapan perancangan, implementasi, dan pengujian sistem telah selesai dilakukan. Kesimpulan diambil untuk menjawab rumusan masalah yang telah ditetapkan berdasarkan hasil pengujian. Tahap terakhir dari penelitian ini adalah muncul saran berkenaan dengan hasil yang telah dicapai untuk memperbaiki kesalahan pada penelitian ini. Saran ini juga bertujuan untuk memberikan pertimbangan pada pengembangan penelitian lebih lanjut.

BAB 4 PERANCANGAN

Bab ini menjelaskan tentang perancangan alur kerja sistem yang akan dibuat guna membuktikan kebenaran studi pustaka pada objek yang diangkat. Perancangan pada penelitian ini dibuat sebagai pedoman dalam implementasi sistem, agar sistem yang nantinya dibuat tidak menyimpang dari pembahasan. Pada bab ini akan dijelaskan lebih detail mengenai deskripsi umum sistem, batasan sistem, alur kerja sistem, perhitungan manual, perancangan tampilan antarmuka dan perancangan pengujian.

4.1 Deskripsi Umum Sistem

Sistem yang dibangun mengimplementasikan *evolution strategies* untuk mengoptimasi regresi linier guna mencari persamaan regresi terbaik dengan jumlah amatan terbanyak pada data. Hasil keluaran sistem merupakan komponen pembentuk persamaan regresi. Kemudian selain menampilkan komponen tersebut sistem juga akan menampilkan seberapa kuat persamaan regresi yang dibentuk dalam ukuran nilai Kriteria Informasi. Di dalam penelitian ini juga digunakan metode optimasi, sehingga sistem juga akan menghasilkan keluaran berupa hasil proses random. Sebelum sistem mengeluarkan hasil optimasi, sistem akan memberikan beberapa pilihan komponen-komponen optimasi pada *evolution strategies* seperti jumlah populasi, rasio offspring dan jumlah generasi.

4.2 Batasan Sistem

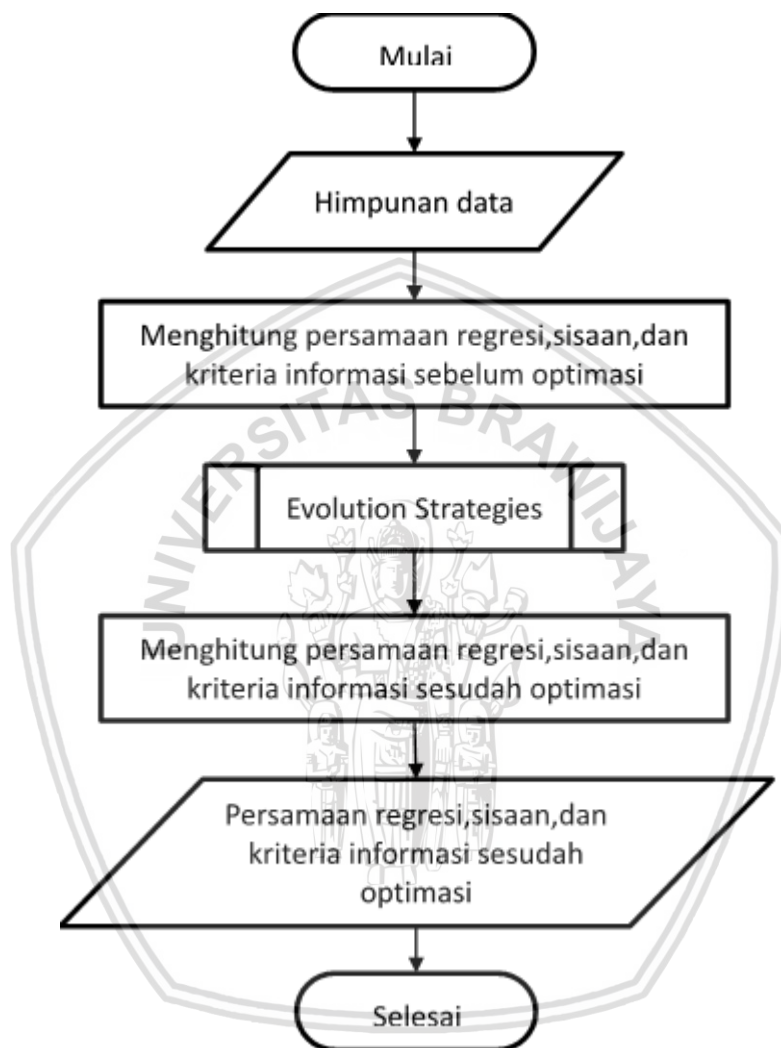
Agar pembahasan penelitian ini lebih mengarah pada permasalahan yang diangkat, maka implementasi sistem harus dibatasi. Batasan dalam implementasi sistem pada penelitian ini antara lain:

1. Data yang telah disimpan akan diproses untuk menghasilkan keluaran berupa variansi, random 1, random 2, data akhir dan fitness akhir.
2. Sebelum pemrosesan data ada parameter yang harus dimasukkan secara manual yaitu jumlah individu, jumlah populasi dan jumlah generasi.
3. Keluaran terakhir dari sistem ini berupa kumpulan fitness dari hasil perhitungan data akhir setelah optimasi yang diambil dari individu terbaik.
4. Sistem yang akan dibangun tidak memberikan batasan maksimal jumlah amatan pada data yang digunakan, namun pada penelitian ini terdapat 30 amatan.
5. Sistem yang akan dibangun memiliki ketentuan jumlah variabel untuk setiap amatan yang terdiri dari 1 variabel bebas dan 1 variabel terikat.

4.3 Alur Kerja Sistem

Alur kerja sistem dibuat sebagai pedoman dalam implementasi agar sistem yang nantinya dibuat tidak menyimpang dari pembahasan. Tujuan lain dari alur kerja sistem yaitu menjelaskan tahapan sistem dalam menjalankan metode yang

digunakan serta mempermudah implementasi. Pada subbab ini alur kerja sistem akan dijelaskan menggunakan diagram alur kerja beserta deskripsinya. Diagram alur kerja merupakan tahapan sistem dalam menjalankan metode yang digambarkan menggunakan simbol sebagai langkah alur kerja dan tanda panah sebagai urutan langkahnya (Software and Systems Engineering Vocabulary, 2008). Diagram alur kerja sistem secara umum ditunjukkan pada Gambar 4.1.



Gambar 4.1 Komponen regresi linier

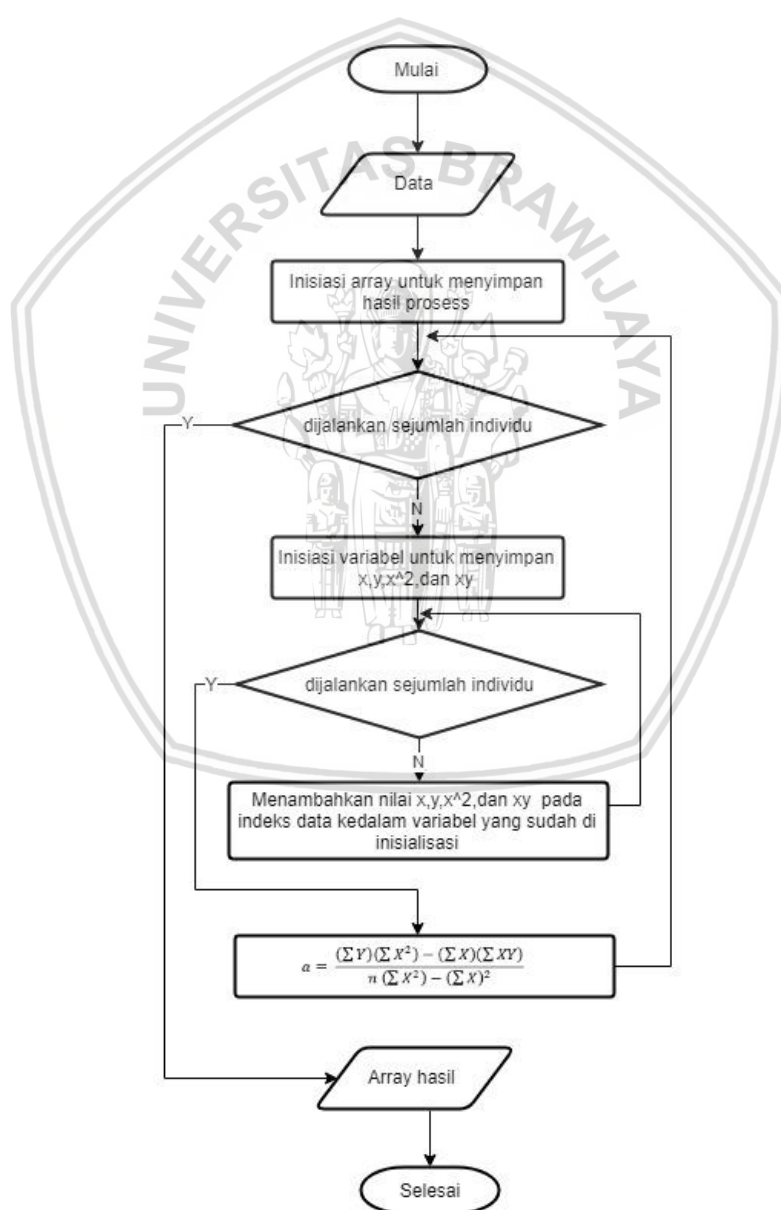
Alur kerja pada Gambar 4.1 dimulai dari mengolah data guna memperoleh komponen regresi linier tanpa optimasi. Setelah proses tersebut data akan dioptimasi menggunakan *evolution strategies* guna mencari amatan mana yang merupakan pencilan. Proses ini akan menghasilkan persamaan regresi beserta nilai kekuatannya dengan dan tanpa optimasi.

4.3.1 Komponen Regresi Linier

Pada sistem ini komponen pada regresi linier akan dibagi menjadi 3 bagian antara lain persamaan regresi, sisaan serta kriteria informasi. Berikut diagram alur kerja pembahasan tentang 3 komponen regresi linier untuk mengolah hasil analisis yang dikelompokkan berdasarkan fungsinya.

4.3.1.1 Proses Pembentukan Konstanta

Proses ini merupakan proses yang dilakukan untuk mencari komponen regresi linier yaitu konstanta (a). Proses ini dilakukan karena konstanta(a) dibutuhkan dalam mencari sisaan dan nilai kriteria informasi. Diagram alur kerja untuk membentuk persamaan regresi linier ditunjukkan pada Gambar 4.2.



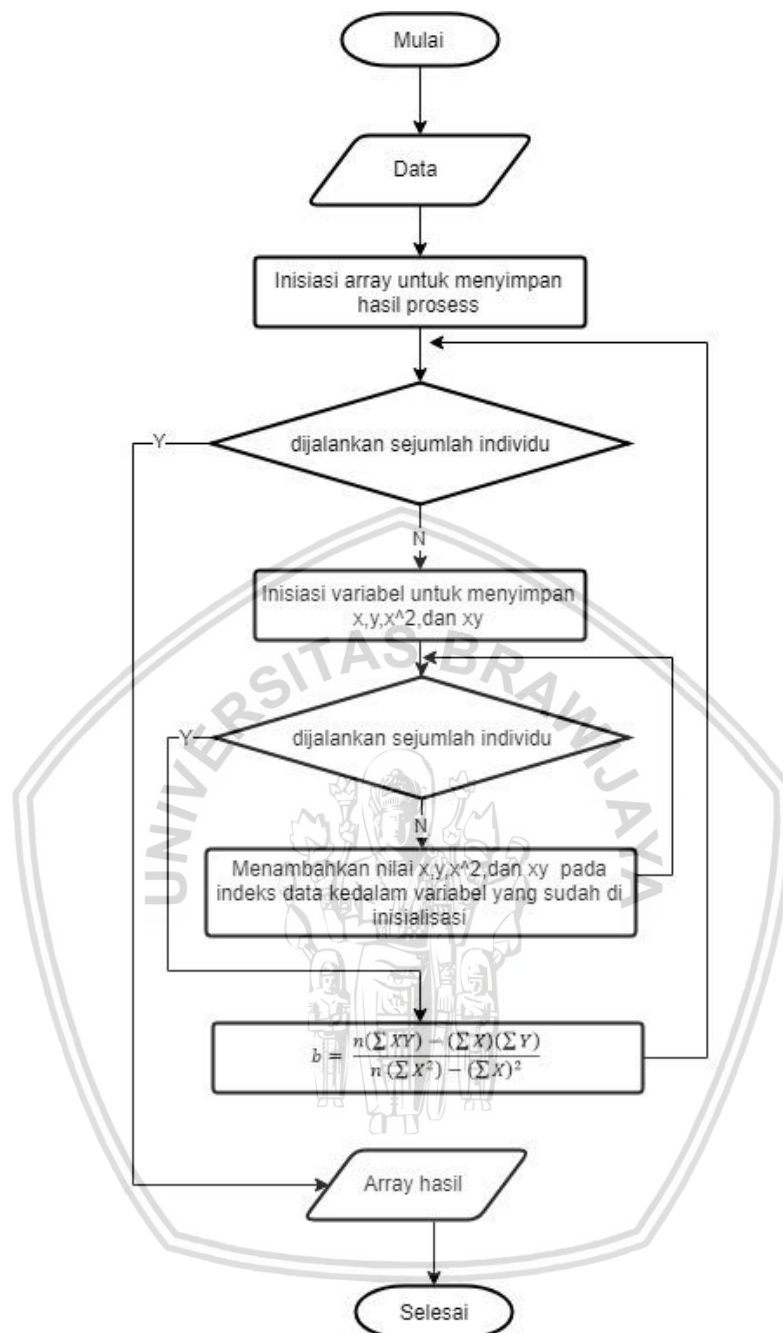
Gambar 4.2 Diagram alur pembentukan konstanta

Alur kerja pada Gambar 4.2 merupakan proses pembentukan konstanta berdasarkan variabel x, y, x^2 , dan xy . Untuk melakukan proses ini tanpa optimasi, parameter yang dimasukkan berupa data awal yaitu x dan y .

4.3.1.2 Proses Pembentukan Kemiringan

Proses ini merupakan proses yang dilakukan untuk mencari komponen regresi linier yaitu kemiringan (b). Proses ini dilakukan karena konstanta(a) dibutuhkan dalam mencari sisaan dan nilai kriteria informasi. Diagram alur kerja untuk membentuk persamaan regresi linier ditunjukkan pada Gambar 4.3.





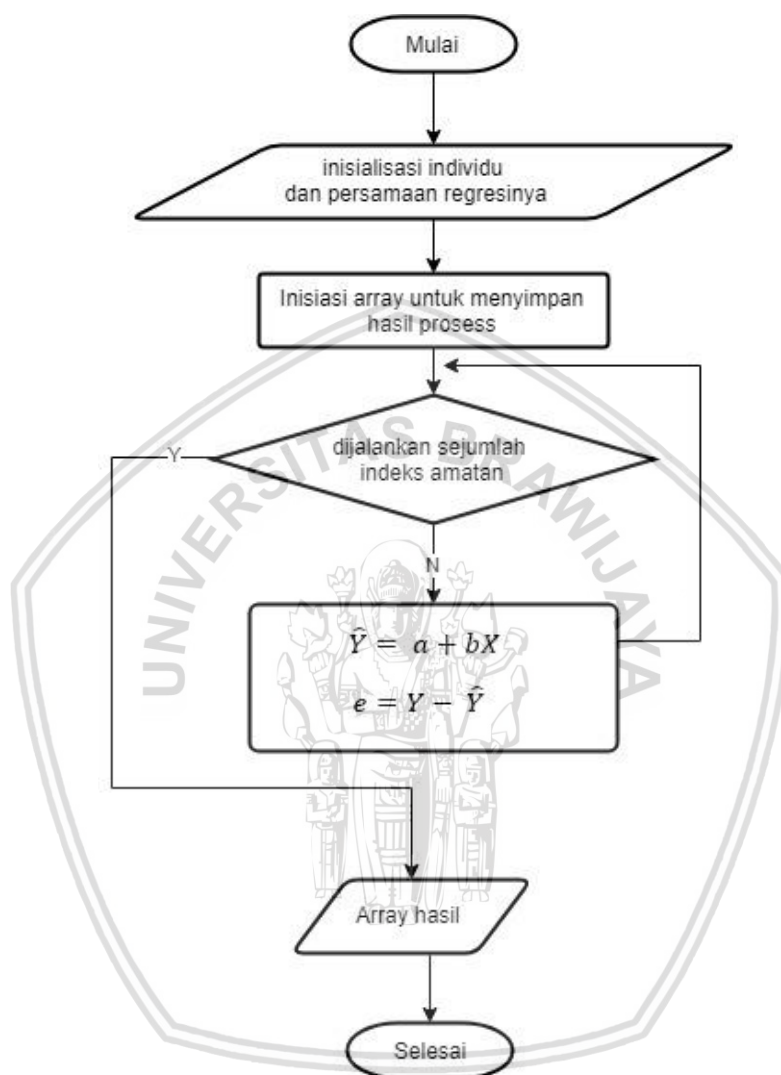
Gambar 4.3 Diagram alur pembentukan kemiringan

Alur kerja pada Gambar 4.3 merupakan proses pembentukan kemiringan berdasarkan variabel x, y, x^2 , dan xy . Untuk melakukan proses ini tanpa optimasi, parameter yang dimasukkan berupa data awal yaitu x dan y .

4.3.1.3 Proses Pembentukan Sisaan

Tujuan utama dari proses ini adalah membentuk sisaan untuk digunakan dalam proses pembentukan kriteria informasi. Tujuan lain dari proses ini yaitu

untuk menampilkan sisaan dan variabel terikat prediksi dari masing-masing amatan dengan dan tanpa optimasi, sehingga proses ini dipisahkan dengan proses pembentukan kriteria informasi. Diagram alur kerja untuk membentuk sisaan dan variabel terikat prediksi pada masing-masing amatan ditunjukkan pada Gambar 4.4.

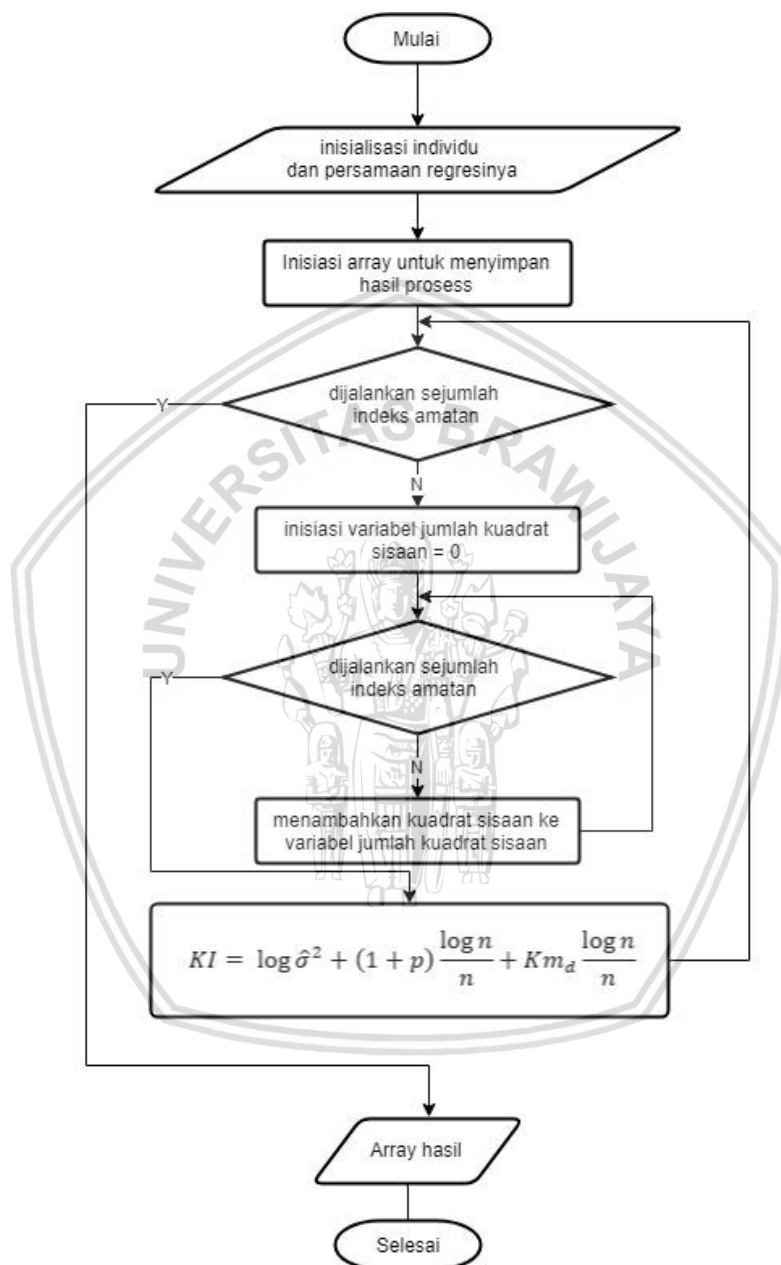


Gambar 4.4 Diagram alur pembentukan sisaan

Alur kerja pada Gambar 4.4 menjelaskan proses dalam mencari nilai sisaan untuk masing-masing amatan. Proses ini membutuhkan persamaan regresi yang dibentuk dari proses sebelumnya yaitu konstanta (a) dan kemiringan (b) untuk dimasukkan ke dalam rumus untuk setiap amatan.

4.3.1.4 Proses Pembentukan Kriteria Informasi

Proses ini bertujuan untuk menghasilkan nilai kriteria informasi pada masing-masing individu dalam populasi yang akan digunakan sebagai *fitness* pada *evolution strategies*. Diagram alur kerja untuk menghitung nilai kriteria informasi untuk masing-masing individu ditunjukkan pada Gambar 4.5.



Gambar 4.5 Diagram alur kerja kriteria informasi

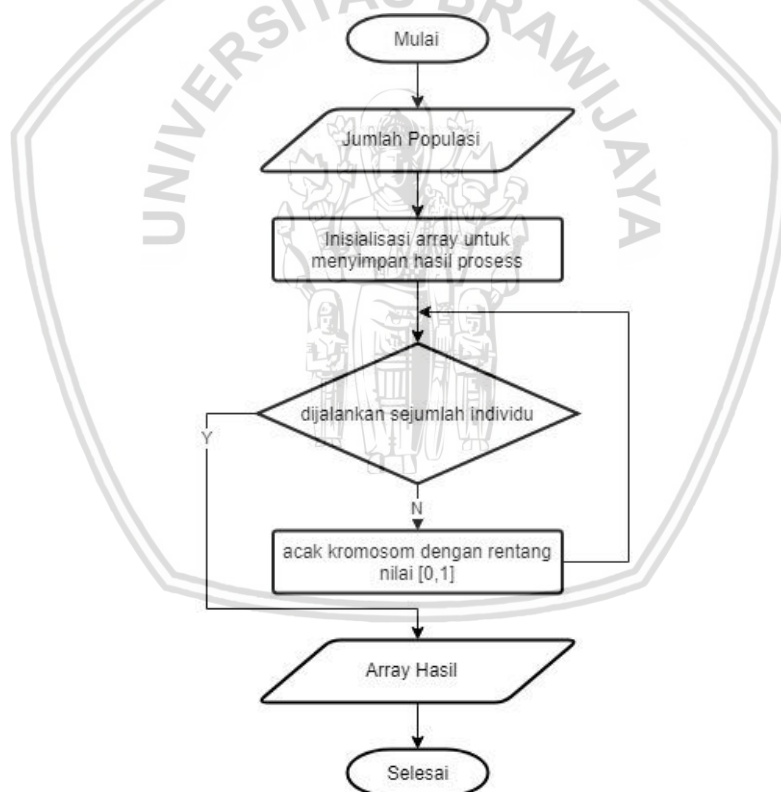
Alur kerja pada Gambar 4.5 menjelaskan proses dalam mencari nilai kriteria informasi pada masing-masing populasi.

4.3.2 Evolution strategies

Proses ini bertujuan untuk melakukan serangkaian proses dalam menjalankan optimasi dengan *evolution strategies*. Serangkaian proses tersebut antara lain inisialisasi, mutasi dan seleksi. Semua proses di atas akan dilakukan sebanyak parameter yang dimasukkan guna merubah susunan individu dalam populasi. Susunan individu dalam populasi dirubah agar menghasilkan susunan individu dengan *fitness* lebih baik, namun seberapa baik *fitness* yang dihasilkan akan tergantung dari parameter yang dimasukkan.

4.3.2.1 Inisialisasi

Inisialisasi populasi awal (μ) merupakan langkah pertama dari algoritma *Evolution Strategies*. Pada tahapan ini, kromosom dari induk akan dibangkitkan secara acak. Setelah representasi kromosomnya terbentuk, maka selanjutnya akan membangkitkan populasi awal yang digunakan sebagai induk pada proses generasi pertama. Gambar 4.6 akan menjabarkan diagram alir dari proses terbentuknya populasi awal.



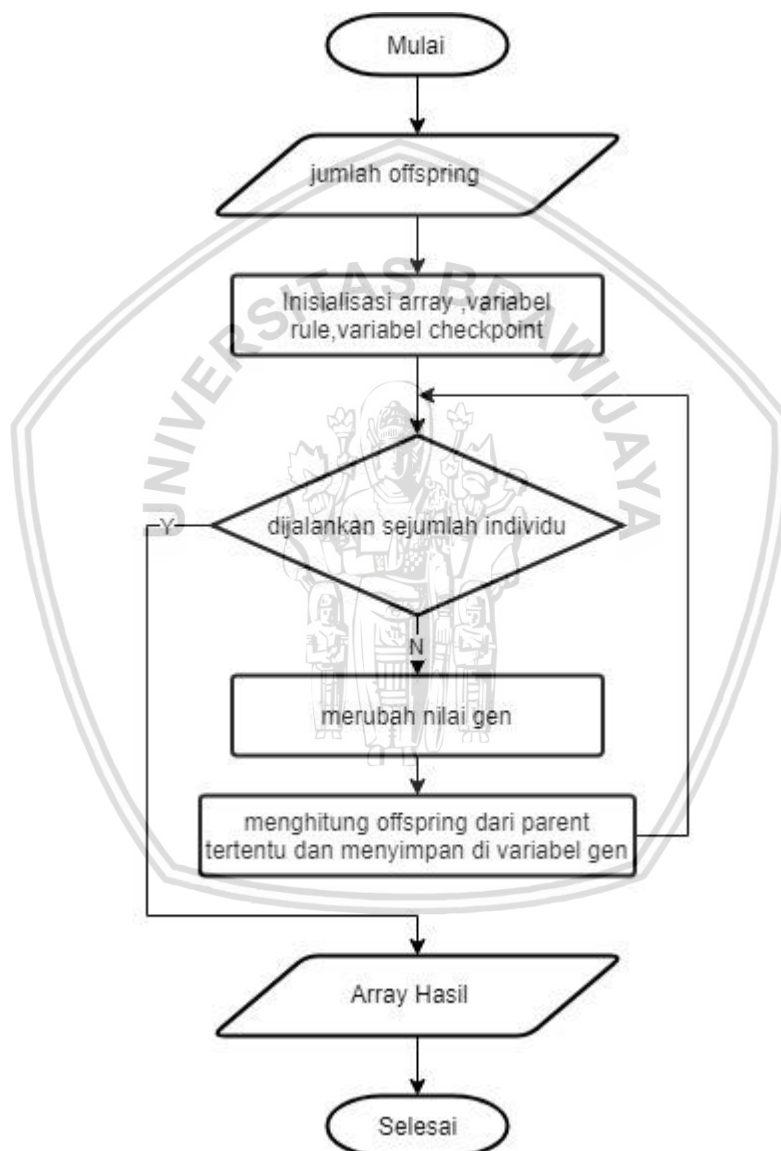
Gambar 4.6 Diagram alur kerja inisialisasi

Alur kerja pada Gambar 4.6 menjelaskan proses Inisialisasi populasi awal dimulai dari menentukan jumlah populasi yang akan dibentuk. Langkah selanjutnya ialah melakukan perulangan sebanyak jumlah populasi tersebut. Pada proses

perulangan terdapat pembangkitan bilangan acak untuk setiap gen dengan rentang nilai 0 sampai 1

4.3.2.2 Mutasi

Proses mutasi pada algoritma *Evolution Strategies* dengan siklus ES ($\mu+\lambda$) tidak menggunakan kromosom induk sebagai induk yang akan menghasilkan *offspring*. Proses ini bertujuan untuk mengubah salah satu gen dari individu terpilih untuk menjadi individu baru. Diagram alur kerja dalam melakukan proses mutasi ditunjukkan pada Gambar 4.7.



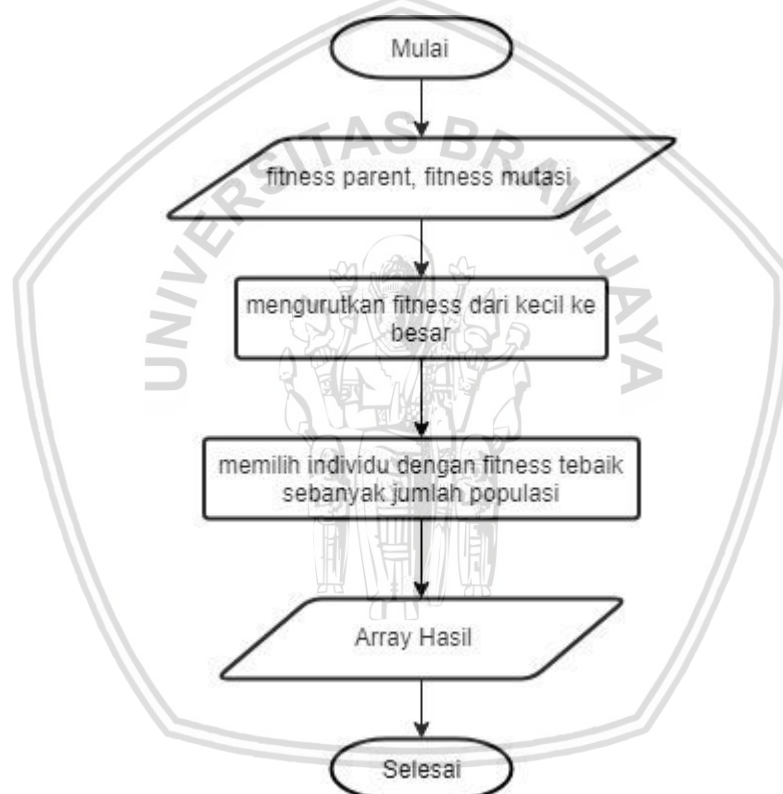
Gambar 4.7 Diagram alur kerja mutasi

Alur kerja pada Gambar 4.7 menjelaskan proses mutasi dimana individu serta salah satu gennya dipilih untuk dirubah nilainya. Hasil akhir dari proses ini

merupakan 1 individu baru hasil mutasi dari individu terpilih untuk ditambahkan dalam populasi.

4.3.2.3 Seleksi

Proses seleksi pada algoritma *Evolution Strategies* menggunakan metode *elitism selection* dimana kromosom dengan nilai *fitness* terbaiklah yang akan terpilih. Penelitian ini menggunakan siklus ES ($\mu + \lambda$), sehingga pada proses seleksinya akan melibatkan kromosom induk dan kromosom *offspring* sebagai kandidat individu yang akan dipilih untuk generasi selanjutnya. Adapun diagram alir proses seleksi akan ditunjukkan pada Gambar 4.8.



Gambar 4.8 Diagram alir kerja mutasi

Alur kerja pada Gambar 4.8 Pada seleksi, terdapat proses pengurutan nilai *fitness* individu yang terlibat dari yang tertinggi sampai yang terendah. Setelah itu, individu dengan nilai *fitness* yang terbaik akan dipilih sebanyak jumlah populasi. individu yang terpilih tersebut akan dijadikan sebagai individu baru untuk generasi selanjutnya.

4.4 Perhitungan Manual

4.4.1 Proses Regresi Linier

Pada penelitian ini perhitungan manual digunakan sebagai pedoman dalam merancang alur kerja dari metode yang digunakan. Perhitungan manual dimulai dari pengolahan data guna membentuk persamaan regresi hingga menghasilkan nilai kriteria informasi untuk masing-masing individu. Berikut data pada penelitian ini dengan kolom kelembapan tanah sebagai variabel bebas dan kolom volume air aktual sebagai variabel terikat yang ditunjukkan pada Tabel 4.1.

Table 4.1 Data

No	Kelembapan	Volume Air Aktual
1	3.75	1.325
2	4.87	2.1
3	4.16	1.4125
4	1.97	0.325
5	1.54	0.1625
6	1.47	0.1
7	3.98	1.65
8	3.47	1.2625
9	2.52	0.4625
10	3.87	1.3625
11	4.43	1.7
12	3.28	1.1625
13	2.11	0.5125
14	3.51	1.3
15	2.94	0.9875
16	1.79	0.1875
17	2.27	0.5625
18	1.67	0.1875
19	2.03	0.425
20	3.34	1.4375
21	3.02	1.2625
22	1.83	0.2125
23	2.41	0.675
24	4.85	2.075
25	2.67	0.7875

26	4.69	1.9
27	4.56	1.8
28	1.72	0.225
29	4.31	1.5125
30	2.86	0.85

Dalam perhitungan nilai kriteria informasi nilai didapat dengan menghitung sisaan (*error*) yang terbentuk dari seluruh amatan. Dari penjelasan di atas meskipun terdapat perbedaan cara memperoleh persamaan namun persamaan yang dibutuhkan tetap sama, sehingga proses awal untuk mengolah data juga sama. Proses awal untuk mengolah data guna memperoleh persamaan regresi ditunjukkan pada Tabel 4.2.

Table 4.2 Proses awal pengolahan data

x(Kelembapan)	y(Volume Air Aktual)	x ²	x*y
3.75	1.325	14.0625	4.96875
4.87	2.1	23.7169	10.227
4.16	1.4125	17.3056	5.876
1.97	0.325	3.8809	0.64025
1.54	0.1625	2.3716	0.25025
1.47	0.1	2.1609	0.147
3.98	1.65	15.8404	6.567
3.47	1.2625	12.0409	4.380875
2.52	0.4625	6.3504	1.1655
3.87	1.3625	14.9769	5.272875
4.43	1.7	19.6249	7.531
3.28	1.1625	10.7584	3.813
2.11	0.5125	4.4521	1.081375
3.51	1.3	12.3201	4.563
2.94	0.9875	8.6436	2.90325
1.79	0.1875	3.2041	0.335625
2.27	0.5625	5.1529	1.276875
1.67	0.1875	2.7889	0.313125
2.03	0.425	4.1209	0.86275
3.34	1.4375	11.1556	4.80125
3.02	1.2625	9.1204	3.81275
1.83	0.2125	3.3489	0.388875
2.41	0.675	5.8081	1.62675
4.85	2.075	23.5225	10.06375
2.67	0.7875	7.1289	2.102625
4.69	1.9	21.9961	8.911
4.56	1.8	20.7936	8.208

1.72	0.225	2.9584	0.387
4.31	1.5125	18.5761	6.518875
2.86	0.85	8.1796	2.431
91.89	29.925	316.3611	111.427375

Tabel 1.2 merupakan proses perhitungan yang harus dilalui untuk mendapatkan persamaan regresi. Proses tersebut membutuhkan jumlah dari masing-masing kolom pada tabel di atas.

Proses perhitungan nilai kriteria informasi dimulai dari membentuk persamaan regresi guna mencari sisaan (*error*) untuk masing-masing amatan. Sisaan ini nantinya akan digunakan untuk menghitung penduga ragam model regresi yang dibutuhkan dalam mencari nilai kriteria informasi. Proses untuk mencari nilai kriteria informasi untuk satu individu ditunjukkan pada Tabel 4.3.

Table 4.3 Proses penjumlahan

x(Kelembapan)	y(Volume Air Aktual)	x ²	x*y
3.75	1.325	14.0625	4.96875
4.87	2.1	23.7169	10.227
4.16	1.4125	17.3056	5.876
1.97	0.325	3.8809	0.64025
1.54	0.1625	2.3716	0.25025
1.47	0.1	2.1609	0.147
3.98	1.65	15.8404	6.567
3.47	1.2625	12.0409	4.380875
2.52	0.4625	6.3504	1.1655
3.87	1.3625	14.9769	5.272875
4.43	1.7	19.6249	7.531
3.28	1.1625	10.7584	3.813
2.11	0.5125	4.4521	1.081375
3.51	1.3	12.3201	4.563
2.94	0.9875	8.6436	2.90325
1.79	0.1875	3.2041	0.335625
2.27	0.5625	5.1529	1.276875
1.67	0.1875	2.7889	0.313125
2.03	0.425	4.1209	0.86275
3.34	1.4375	11.1556	4.80125
3.02	1.2625	9.1204	3.81275
1.83	0.2125	3.3489	0.388875
2.41	0.675	5.8081	1.62675
4.85	2.075	23.5225	10.06375
2.67	0.7875	7.1289	2.102625

4.69	1.9	21.9961	8.911
4.56	1.8	20.7936	8.208
1.72	0.225	2.9584	0.387
4.31	1.5125	18.5761	6.518875
2.86	0.85	8.1796	2.431
91.89	29.925	316.3611	111.427375

Seperti yang sudah dijelaskan kriteria informasi membutuhkan persamaan regresi dari seluruh amatan sehingga tidak perlu memperhitungkan susunan kromosom dalam menjumlahkan kolom pada tabel di atas. Setelah masing-masing kolom pada tabel di atas dijumlahkan maka proses selanjutnya adalah persamaan regresi yang terdiri dari konstanta dan kemiringan. Berikut cara untuk menghitung persamaan regresi.

1. Hitung nilai konstanta (a).

$$a = \frac{(\sum Y)(\sum X^2) - (\sum X)(\sum XY)}{n(\sum X^2) - (\sum X)^2}$$

$$a = \frac{(29.925)(316.3611) - (91.89)(111.427375)}{30(316.3611) - (91.89)^2}$$

$$a = -0.737259477$$

2. Hitung nilai kemiringan (b).

$$b = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum X^2) - (\sum X)^2}$$

$$b = \frac{30(111.427375) - (91.89)(29.925)}{30(316.3611) - (91.89)^2}$$

$$b = 0.566359607$$

Setelah konstanta dan kemiringan didapatkan maka proses selanjutnya adalah menghitung variabel terikat prediksi. Variabel terikat prediksi didapat dengan memasukkan variabel bebas pada persamaan regresi. Proses perhitungan penduga ragam model regresi membutuhkan nilai sisaan dari selisih antara variabel terikat pada tabel dengan variabel terikat prediksi. Proses untuk mendapatkan nilai sisaan untuk masing-masing amatan ditunjukkan pada Tabel 4.4.

Table 4.4 Proses perhitungan sisaan

Y'(Volume Air Prediksi)	e(Sisaan Error)
1.38658905	-0.06158905
2.02091181	0.07908819
1.618796489	-0.206296489
0.378468949	-0.053468949

0.134934318	0.027565682
0.095289146	0.004710854
1.51685176	0.13314824
1.22800836	0.03449164
0.689966733	-0.227466733
1.454552203	-0.092052203
1.771713583	-0.071713583
1.120400035	0.042099965
0.457759294	0.054740706
1.250662744	0.049337256
0.927837768	0.059662232
0.27652422	-0.08902422
0.548376832	0.014123168
0.208561067	-0.021061067
0.412450526	0.012549474
1.154381611	0.283118389
0.973146537	0.289353463
0.299178604	-0.086678604
0.627667177	0.047332823
2.009584618	0.065415382
0.774920674	0.012579326
1.918967081	-0.018967081
1.845340332	-0.045340332
0.236879048	-0.011879048
1.70375043	-0.19125043
0.882529	-0.032529

Tabel 1.4 merupakan variabel terikat prediksi dan sisaan untuk setiap amatan. Berikut cara menghitung variabel terikat prediksi dan sisaan dengan contoh amatan ke-1 menggunakan persamaan regresi dari proses sebelumnya.

1. Menghitung variabel terikat prediksi.

$$\hat{Y} = a + bX$$

$$\hat{Y} = -0.737259477 + 0.566359607 * 3.75$$

$$\hat{Y} = 1.386589$$

2. Menghitung sisaan.

$$e = Y - \hat{Y}$$

$$e = 1.325 - 1.386589$$

$$e = -0.06159$$

Penduga ragam model regresi akan digunakan untuk menghitung nilai kriteria informasi, yang mana digunakan sebagai *fitness* pada *evolution strategies*. Proses perhitungan penduga ragam model regresi ditunjukkan pada Tabel 4.5.

Table 4.5 Proses penjumlahan sisaan kuadrat pada individu

e(Sisaan Error)	e ² (Sisaan Error Kuadrat)
-0.06158905	0.003793
0.07908819	0.006255
-0.206296489	0.042558
-0.053468949	0.002859
0.027565682	0.000760
0.004710854	0.000022
0.13314824	0.017728
0.03449164	0.001190
-0.227466733	0.051741
-0.092052203	0.008474
-0.071713583	0.005143
0.042099965	0.001772
0.054740706	0.002997
0.049337256	0.002434
0.059662232	0.003560
-0.08902422	0.007925
0.014123168	0.000199
-0.021061067	0.000444
0.012549474	0.000157
0.283118389	0.080156
0.289353463	0.083725
-0.086678604	0.007513
0.047332823	0.002240
0.065415382	0.004279
0.012579326	0.000158
-0.018967081	0.000360
-0.045340332	0.002056
-0.011879048	0.000141
-0.19125043	0.036577
-0.032529	0.001058
-0.0000000000001207	0.378275509

Tabel 4.5 merupakan proses untuk mencari jumlah sisaan kuadrat guna mencari penduga ragam model regresi. Kolom individu merupakan contoh susunan gen yang akan dihitung nilai kriteria informasinya. Sisaan kuadrat yang dijumlahkan hanya untuk amatan dengan gen bernilai 1 saja. Setelah

mendapatkan jumlah sisaan kuadrat untuk individu dengan susunan gen pada tabel di atas maka proses selanjutnya adalah menghitung penduga ragam model regresi untuk mendapatkan nilai kriteria informasi. Berikut proses untuk menghitung penduga ragam model regresi dan nilai kriteria informasi.

1. Menghitung penduga ragam model regresi dengan cara membagi sisaan kuadrat dengan jumlah data pencilan. Sebuah data dianggap pencilan juga nilai

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n-m_d} (e_i^2)}{n - m_d}$$

$$\hat{\sigma}^2 = \frac{0.06579}{30 - 6}$$

$$\hat{\sigma}^2 = 0.00274123$$

2. Menghitung kriteria informasi.

$$KI = \log \hat{\sigma}^2 + (1 + p) \frac{\log n}{n} + K m_d \frac{\log n}{n}$$

$$KI = \log 0.00274123 + (1 + 1) \frac{\log 30}{30} + 1.5 * 6 \frac{\log 30}{30}$$

$$KI = -2.020443374$$

Proses di atas merupakan hasil akhir dari perhitungan nilai kriteria informasi. Berdasarkan proses di atas, nilai kriteria informasi akan berbeda untuk setiap susunan gen pada suatu individu meskipun dalam proses pencarian persamaan regresinya tanpa melibatkan susunan gen.

4.4.2 Proses Representasi Kromosom

Pada Evolution Strategies proses pertama yang dilakukan adalah representasi kromosom. Dan pada penelitian ini representasi kromosom yang digunakan adalah dengan bentuk bilangan decimal dimana rentang nilai yang dimiliki adalah antara 0 sampai 1 .

Algoritma *Evolution Strategies* memiliki mekanisme *self-adaptation* dimana perbandingan antara nilai *fitness* induk dan nilai *fitness offspring* digunakan sebagai acuan untuk merubah nilai secara adaptif. Nilai sigma (σ) menandai adanya mekanisme *self-adaptation* dimana nilai sigma (σ) adalah bilangan acak dengan rentang nilai [0,1]. Setiap gen akan memiliki satu nilai sigma (σ). Tabel 4.6 akan menunjukkan contoh representasi kromosom untuk optimasi populasi ke-1 dari induk.

Table 4.6 Representasi Kromosom

Populasi 1	Populasi 11	Populasi 12
0.310198309	0.569099412	0.522572359
0.80899983	0.542143502	0.971832856
0.565634384	0.351296923	0.013948505
0.037667389	0.062080898	0.366931046
0.397697766	0.49921101	0.198729928
0.003793666	0.07960085	0.791768247
0.703663882	0.581325516	0.124978313
0.293525244	0.96163795	0.523655549
0.949653682	0.470101447	0.880563761
0.918639522	0.353714747	0.108464305
0.468272053	0.721334339	0.636633474
0.599126377	0.170360748	0.141552827
0.771201994	0.929455124	0.102393361
0.82189219	0.08948378	0.521686643
0.793212226	0.468645655	0.51538691
0.609811056	0.673675089	0.426896362
0.347362182	0.479431479	0.521374482
0.10338653	0.88440905	0.938273118
0.655792717	0.507980481	0.5854334
0.712144678	0.390324621	0.195193841
0.358212212	0.655059131	0.010986005
0.768343067	0.425020369	0.217345984
0.058141024	0.347783753	0.02225504
0.044801002	0.997334191	0.095738491
0.674836453	0.579665579	0.737874105
0.652737997	0.031420695	0.429295951
0.942476889	0.4879402	0.402398152
0.717655345	0.119629904	0.602258385
0.217551073	0.027552251	0.814136284
0.111298052	0.415956158	0.136725212

Perancangan Tampilan Antarmuka

Perancangan tampilan antarmuka dibuat sebagai pedoman dalam implementasi tampilan antarmuka pada sistem yang nantinya akan dibuat. Perancangan tampilan antarmuka terdiri dari 2 bagian yaitu bagian input data untuk parameter optimasi dan bagian print data hasil optimasi. Gambar rancangan antarmuka sistem ditunjukkan pada Gambar 4.10

1
2
3
4

Gambar 4.10 Rancangan tampilan antarmuka

Gambar 4.10 merupakan rancangan tampilan antarmuka. Berikut penjelasan untuk setiap bagian dari antarmuka.

1. Bagian nomor 1 merupakan kepala yang memuat nama sistem.
2. Bagian nomor 2 berfungsi untuk menginputkan data dan parameter parameter optimasi.
3. Bagian nomor 3 memuat kolom untuk memilih proses data yang ditampilkan.
4. Bagian nomor 4 merupakan kolom untuk menampilkan hasil proses data.

4.5 Perancangan Pengujian dan Analisis

Perancangan pengujian dilakukan untuk menguji parameter-parameter yang digunakan dalam algoritma *Evolution Strategies*. Adapun tujuan dilakukannya pengujian ini ialah untuk mendapatkan nilai parameter yang terbaik yang dapat mendukung algoritma *Evolution strategies* untuk dapat menyelesaikan permasalahan optimasi biaya bahan menu makanan bagi penderita penyakit jantung. Parameter-parameter yang akan diuji ialah sebagai berikut:

1. Jumlah populasi (μ)
2. Jumlah *offspring* (λ)
3. Banyak generasi

4.5.1 Perancangan Pengujian Populasi

Pengujian jumlah populasi dilakukan untuk mengetahui pengaruh banyaknya populasi yang terlibat terhadap nilai *fitness* yang dihasilkan sebagai solusi yang optimal. Pengujian jumlah populasi akan dilakukan sebanyak 10 kali untuk setiap jumlah populasi yang akan diuji. Selain jumlah populasi, parameter lain yang digunakan pada pengujian jumlah populasi ini adalah sebagai berikut:

Jumlah generasi : 10

Jumlah *offspring* (λ) : $0,5 \mu$

Table 4.6 Pengujian Populasi

Jumlah Populasi	Fitness percobaan ke-				rata-rata fitness
	1	2	10	
10					
20					
30					
...					
100					

4.5.2 Perancangan Pengujian *Offspring*

Pengujian jumlah *offspring* dilakukan untuk mengetahui pengaruh banyaknya *offspring* yang akan dihasilkan pada proses reproduksi terhadap nilai *fitness* yang didapatkan sebagai solusi yang optimal. Pengujian jumlah *offspring* akan dilakukan sebanyak 10 kali untuk setiap jumlah *offspring* yang akan diuji. Selain jumlah *offspring*, parameter lain yang digunakan (tidak berubah) pada pengujian jumlah *offspring* ini adalah sebagai berikut:

Jumlah populasi (μ) : 40

Jumlah *offspring* (λ) : $0.1\mu - 1 \mu$

Table 4.7 Pengujian *Offspring*

Jumlah Offspring	Fitness percobaan ke-				rata-rata fitness
	1	2	10	
0.1					
0.2					
0.3					
...					
1					

4.5.3 Perancangan Pengujian Generasi

Pengujian banyak generasi dilakukan untuk mengetahui pengaruh banyaknya generasi terhadap nilai *fitness* yang didapatkan sebagai solusi yang optimal. Pengujian banyak generasi akan dilakukan sebanyak 10 kali untuk setiap jumlah generasi yang akan diuji. Selain banyak generasi, parameter lain yang digunakan (tidak berubah) pada pengujian banyak generasi ini adalah sebagai berikut:

Jumlah populasi (μ) : 40

Jumlah *offspring* (λ) : 0.4μ

Table 4.8 Pengujian Generasi

Jumlah Generasi	Fitness percobaan ke-				rata-rata fitness
	1	2	10	
10					
20					
30					
...					
100					



BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang implementasi sistem berdasarkan metode yang digunakan pada penelitian ini. Implementasi pada penelitian ini bertujuan untuk membuat sistem yang lebih dinamis sehingga dapat mengolah data sesuai dengan metode yang digunakan. Pada bab ini akan dibahas secara detail tentang sistem yang dibuat beserta tampilan antarmuka dan perangkat yang digunakan.

5.1 Perangkat Implementasi

Analisis kebutuhan pada penelitian ini dilakukan untuk menerangkan perangkat yang digunakan dalam perhitungan manual serta implementasi sistem. Daftar perangkat yang digunakan antara lain:

1. Perangkat keras
 - a. Laptop
 - Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz (4CPUs)
 - RAM 4 GB
 - Harddisk kapasitas 500 GB
 - Monitor 14 inch
 - b. Printer
2. Perangkat lunak
 - a. Netbeans
 - b. Sublime Text
 - c. Google Chrome
 - d. Microsoft Excel 2013

Berikut perangkat yang digunakan untuk perhitungan manual dan implementasi sistem pada penelitian ini. Selain perhitungan manual serta implementasi perangkat di atas juga digunakan oleh peneliti untuk melakukan analisis dan pengujian.

5.2 Implementasi Proses *Evolution strategies*

Secara umum proses ini dilakukan untuk merubah susunan populasi yang sudah ada agar terbentuk solusi-solusi baru. Susunan populasi ini akan dirubah melalui proses mutasi kemudian diseleksi. Hasil akhir dari proses ini adalah populasi baru yang telah dirubah sesuai dengan parameter yang dimasukkan.

1	<code>public ES(String nameFile, int maxPopulasi, double offspring,</code>
2	<code>int iteration) {</code>
3	<code> file = new Array(nameFile);</code>
4	<code> this.maxPopulasi = maxPopulasi;</code>
	<code> maxGen=(int) (offspring *maxPopulasi);</code>

5	isi = new double[maxPopulasi + maxPopulasi * maxGen][file.isi.length][file.isi[0].length];
6	populasi(this.maxPopulasi);
7	variansi = variansi();
8	random1 = random(maxGen);
9	random2 = random(maxGen);
10	int stoper = 0;
11	int f = 0;
12	fitnessC = new double[maxPopulasi];
13	do {
14
15	} while (f < iteration);
16	}

Kode 5.1 Kelas ES

Kode 5.1 adalah kelas algoritme *Evolution Strategies* (ES). Berikut adalah penjelasan kode 5.1 :

1. Baris nomor 1, inialisasi kelas ES. Kelas ES memiliki parameter nameFile (tipe datanya String), maxPopulasi (tipe datanya Integer), offspring (tipe datanya Double) dan iteration (tipe datanya Integer)
2. Baris nomor 2, mengambil data dari file yang dipilih.
3. Baris nomor 3, menyimpan nilai populasi maksimal yang dimasukkan.
4. Baris nomor 4, menyimpan nilai generasi maksimal. Ukuran generasi maksimal didapat dari perkalian antara offspring dan populasi maksimal.
5. Baris nomor 5, membuat array 3 dimensi untuk menyimpan hasil perhitungan ES.
6. Baris nomor 6, memanggil method populasi dengan parameter populasi maksimum. Hasil dari method populasi adalah *parent*.
7. Baris nomor 7, memanggil method variansi.
8. Baris nomor 8-9, memanggil method random dengan parameter generasi maksimal. Method random digunakan untuk membuat membuat *child*.
9. Baris ke 10-12, inialisasi variabel stoper, f dan array fitness.
10. Baris ke 13-15, perulangan ES dilakukan selama nilai f kurang dari iteration.

5.2.1 Implementasi Inialisasi Populasi

Proses ini adalah tahapan pertama yang dilakukan dalam siklus algoritma *Evolution strategies*. Proses inialisasi populasi awal dilakukan dengan cara

mengacak nilai setiap gen yang membentuk kromosom. Jumlah populasi awal ditentukan oleh user saat parameter diisi pada halaman antarmuka. Proses implementasi inisialisasi populasi awal akan dijabarkan pada Kode 5.2 berikut.

1	public void populasi(int maxPopulasi) {
2	this.populasi = new
3	double[maxPopulasi][file.isi.length];
4	for (int i = 0; i <populasi.length; i++) {
5	for (int j = 0; j <populasi[0].length; j++) {
6	populasi[i][j] = Math.random();
7	}
8	}

Kode 5.2 Inisialisasi Populasi Awal

Kode 5.2 bertujuan untuk membuat parent yang akan digunakan dalam algoritma ES. Berikut adalah penjelasan Kode 5.2

1. Baris nomor 2 terdapat deklarasi array dengan tipe double untuk tempat penyimpanan nilai setiap populasi.
2. Baris nomor 3-5 terdapat kondisi perulangan proses random untuk setiap populasi.

5.2.2 Implementasi Perhitungan Variansi

Dalam ES, variansi ditentukan secara acak dalam *range* 0 sampai 1. Dimensi variansi ukurannya sama dengan data yang diisi pada halaman antarmuka. Kode 5.3 menjabarkan proses pembentukan variansi.

1	public double[][] variansi() {
2	double[][] variansi = new
3	double[isi.length][file.isi.length];
4	int check = 0;
5	for (int i = 0; i < variansi.length; i++) {
6	if (i < maxPopulasi) {
7	for (int j = 0; j < variansi[0].length; j++) {
8	variansi[i][j] = Math.random();
9	}
10	} else {
11	for (int j = 0; j < variansi[0].length; j++) {
12	variansi[i][j] = variansi[check][j];
13	}
14	if (check < maxPopulasi) {
15	check++;
16	} else {
	check = 0;

17	}
18	}
19	}
20	return variansi;
21	}

Kode 5.3. Implementasi Perhitungan Variansi

Penjelasan Kode 5.3 adalah sebagai berikut :

1. Baris 1, inisialisasi method variansi. Tipe method variansi adalah double. Oleh karena itu, method variansi harus memiliki nilai kembalian.
2. Baris 2, inisialisasi array variansi.
3. Baris 3, inisialisasi variabel check.
4. Baris 4-19, pengisian array variansi.
5. Baris 4-8, pengisian array variansi *parent* secara acak.
6. Baris 9-18, pengisian array variansi *child* sesuai variansi *parent*.

5.2.3 Implementasi Pembentukan Nilai *N*

Sebelum melakukan reproduksi dalam ES, perlu dibangkitkan nilai *N*. *N* memiliki rentang nilai antara 0 sampai 1. Kode 5.4 menjelaskan bagaimana nilai *N* ditentukan.

1	public double[][] N(int gen) {
2	double[][] N = new double[maxPopulasi * gen][file.isi.length];
3	for (int i = 0; i < N.length; i++) {
4	for (int j = 0; j < N[0].length; j++) {
5	N[i][j] = Math.pow(-2 * 2.71828182845904 * Math.log(random1[i][j]), 0.5) * Math.sin(2) * Math.PI * random2[i][j];
6	}
7	}
8	return N;
9	}

Kode 5.4. Pembentukan Nilai *N*

Penjelasan Kode 5.4 adalah sebagai berikut:

1. Baris 1, inisialisasi method *N*.
2. Baris 2, inisialisasi variabel *N*.
3. Baris 3-7, menghitung nilai *N*.

5.2.4 Implementasi Mutasi

Proses ini bertujuan untuk menghasilkan individu baru yang mirip dengan induk terpilih. Kode 5.5 menjelaskan bagaimana mutasi dilakukan.

1	public double[][] gen(int check) {
2	double[][] gen = new
	double[N.length][file.isi.length];
3	int rule = 1;
4	int checkpoint = 0;
5	for (int i = 0; i < N.length; i++) {
6	if (i == rule * check - 1) {
7	rule++;
8	checkpoint = 0;
9	}
10	for (int j = 0; j < N[0].length; j++) {
11	gen[i][j] = populasi[checkpoint][j] +
12	variansi[checkpoint][j] * N[i][j];
13	checkpoint++;
14	}
15	return gen;
16	}

Kode 5.5 Proses Mutasi

Kode 5.5 merupakan proses untuk menghasilkan gen baru. Berikut penjelasan langkah-langkah untuk kode tersebut.

1. Baris ke 1, inisialisasi method gen.
2. Baris ke 2-4, inisialisasi array gen, variabel rule dan variabel checkpoint.
3. Baris ke 5-14, pengisian array gen
4. Baris ke 6-9, memastikan setiap *parent* memiliki *child* dengan jumlah yang sudah ditentukan.
5. Baris nomor 10-12, menghasilkan *child* dari *parent* tertentu dan menyimpannya di variabel gen.

5.2.5 Implementasi Perhitungan Fitness

Setelah *parent* dan *child* dihasilkan, proses selanjutnya adalah menghitung fitness. Kode 5.6 menjelaskan proses perhitungan fitness.

1	for (int i = 0; i < isi.length; i++) {
2	R = new regresLinier(isi[i]);
3	fitness[i] = R.kriteriaInformasi;
4	}

Kode 5.6 Proses Perhitungan Fitness

Kode 5.6 menjelaskan bagaimana fitness ES dihitung. Pada tahap ini, ES akan membuat objek R dari kelas regresiLinier. Setelah itu, nilai fitness akan disimpan di variabel fitness. Penjelasan Kode 5.6 adalah sebagai berikut:

1. Baris 1-4, perulangan untuk menghitung nilai fitness.
2. Baris 2, membuat objek dari kelas regresiLinier.
3. Baris 3, menyimpan hasil perhitungan fitness.

5.2.6 Implementasi Seleksi

Tahap terakhir dari setiap iterasi ES adalah seleksi. Proses ini dijelaskan pada Kode 5.7.

1	for (int i = 0; i < isi.length; i++) {
2	for (int j = 0; j < isi.length; j++) {
3	if (fitness[i] < fitness[j]) {
4	double temp[] = isi[i];
5	isi[i] = isi[j];
6	isi[j] = temp;
7	double temp1 = fitness[i];
8	fitness[i] = fitness[j];
9	fitness[j] = temp1;
10	double temp3[] = variansi[i];
11	variansi[i] = variansi[j];
12	variansi[j] = temp3;
13	}
14	}
15	}

Kode 5.7 Proses Seleksi

Penjelasan Kode 5.7 adalah sebagai berikut:

1. Baris 3-13, memastikan fitness disimpan dengan urutan kecil ke besar.
2. Baris 4-6, mengganti variabel isi ke-j dengan variabel isi ke-i. i memiliki fitness lebih kecil dari j.
3. Baris 7-9, mengganti fitness ke-j dengan fitness ke-i.
4. Baris 10-12, mengganti variansi ke-j dengan variansi ke-i.

5.3 Implementasi Proses Regresi Linier

Implementasi regresi linier akan membahas secara detail seluruh proses yang berkaitan dengan metode regresi linier. Implementasi regresi linier akan dibagi

menjadi 3 bagian: pembentukan persamaan regresi, perhitungan sisaan, dan perhitungan kriteria informasi.

5.3.1 Implementasi Pembentukan Persamaan Regresi

Secara umum proses ini akan membentuk persamaan regresi dari data berdasarkan susunan gen masing-masing individu dalam populasi. Untuk mencari nilai kriteria informasi yang membutuhkan persamaan regresi tanpa terikat susunan gen akan dimasukkan 1 individu dalam populasi dengan susunan gen keseluruhan 1 sehingga menghasilkan persamaan regresi dengan seluruh amatan. Hasil akhir dari proses ini merupakan nilai konstanta dan kemiringan dalam bentuk *array* sebanyak individu.

```

1 public regresLinier(double data[][]) {
2     this.data = data;
3     n = 0;
4     for (int i = 0; i < data.length; i++) {
5         if (data[i][0] != 0) {
6             n++;
7         }
8     }
9     dataX = dataX();
10    dataY = dataY();
11    xKuadrat = xKuadrat(dataX);
12    xXy = xKaliy(dataX, dataY);
13    averageY = averageY(dataY);
14    kuadratTotal = kuadratTotal(dataY);
15    sumX = sum(dataX);
16    sumY = sum(dataY);
17    sumXY = sum(xXy);
18    sumX2 = sum(xKuadrat);
19    sumKt = sum(kuadratTotal);
20
21    a = a();
22    b = b();
23    yAksen = yAksen(dataX);
24    e = e(dataY);
25    e2 = eKuadrat(e);
26    kuadratRegresi = kuadratRegresi(yAksen(dataX));
27    sumKr = sum(kuadratRegresi);

```

28	sume2 = sum(e2);
29	r2 = rKuadrat();
30	ragamKuadrat();
31	kriteriaInformasi(); }

Kode 5.1 Pembentukan persamaan regresi

Kode 5.1 merupakan proses untuk mencari persamaan regresi. Berikut penjelasan langkah-langkah untuk kode tersebut.

1. Baris nomor 1-3 terdapat kepala *method* dengan parameter berupa array 2 dimensi yang akan dicari persamaan regresi dan rata-rata variabel terikatnya untuk masing-masing individu.
2. Baris nomor 4-6 terdapat kondisi untuk proses pengulangan pada untuk setiap data yang terdapat pada array.
3. Baris nomor 9-30 terdapat proses perhitungan untuk parameter parameter yang digunakan pada regresi.

5.3.2 Implementasi Pembentukan Sisaan

Proses ini bertujuan untuk mencari sisaan pada masing-masing amatan guna mencari nilai kriteria informasi. Proses ini dibedakan dengan proses untuk mencari hasil akhir kriteria informasi dikarenakan sisaan akan ditampilkan terlebih dahulu sebelum diolah. Hasil akhir dari proses ini berupa nilai sisaan dalam bentuk *array* sejumlah amatan.

1	public double a() {
2	// System.out.println(sumX2 + " ----- " + sumX);
3	// System.out.println("(" + sumY + " * " + sumX2 + " - " + sumX + " * " + sumXY + ") / (" + n + " * (" + sumX2 + ") - Math.pow(" + sumX + ", 2))");
4	return (sumY * sumX2 - sumX * sumXY) / (n * (sumX2) - Math.pow(sumX, 2));
5	}
6	
7	public double b() {
8	return (n * sumXY - sumX * sumY) / (n * (sumX2) - Math.pow(sumX, 2));
9	}
10	
11	public double[] yAksen(double[] dataX) {
12	double yAksen[] = new double[dataX.length];
13	for (int i = 0; i < dataX.length; i++) {
14	yAksen[i] = a + b * dataX[i];
15	}
16	return yAksen;

17	}
18	
19	public double[] e(double[] dataY) {
20	double e[] = new double[dataY.length];
21	for (int i = 0; i < dataY.length; i++) {
22	e[i] = dataY[i] - yAksen[i];
23	}
24	return e;

Kode 5.2 Pembentukan sisaan

Kode 5.2 merupakan proses untuk membentuk *array* sisaan. Berikut penjelasan langkah-langkah untuk kode tersebut.

1. Baris nomor 1 -5 terdapat proses perhitungan konstanta.
2. Baris nomor 7-8 terdapat proses perhitungan kemiringan.
3. Baris nomor 11-17 terdapat kondisi untuk menghasilkan nilai y' dimana nilai konstanta(a) dijumlahkan dengan hasil perkalian kemiringan(b) dengan x. yang diulang pada setiap amatan dengan gen bernilai 1 sesuai populasi di parameter.
4. Baris nomor 19-24 terdapat proses perhitungan sisaan dimana nilai data asli (Y) dikurangi y' dan proses ini diulang pada setiap populasi amatan pada parameter.

5.3.3 Implementasi Perhitungan Nilai Kriteria Informasi

Proses ini bertujuan untuk menghitung penduga ragam model regresi dan kriteria informasi berdasarkan sisaan yang sudah didapatkan dan susunan gen masing-masing individu dalam populasi. Setiap individu dalam populasi yang diolah dalam proses ini akan memiliki nilai kriteria informasi, sehingga hasil dari proses ini akan berupa *array* sebanyak individu.

1	public double kriteriaInformasi() {
2	kriteriaInformasi=Math.log (ragamKuadrat) + (1+1) *Math.log (n) /
3	data.length+1.5* (data.length-n) *Math.log (n) /data.length;
4	return kriteriaInformasi;
5	}

Kode 5.3 Perhitungan nilai kriteria informasi

Kode 5.3 merupakan proses untuk mendapatkan nilai kriteria informasi untuk setiap individu dalam populasi. Berikut penjelasan langkah-langkah untuk kode tersebut.

1. Baris nomor 1 terdapat kepala *method* dengan tipe data double.
2. Baris nomor 2 terdapat proses perhitungan kriteria informasi dimana proses ini diulang pada setiap populasi pada parameter

5.4 Implementasi Proses *Evolution strategies*

Secara umum proses ini dilakukan untuk merubah susunan populasi yang sudah ada agar terbentuk solusi-solusi baru. Susunan populasi ini akan dirubah melalui proses persilangan dan mutasi kemudian diseleksi. Hasil akhir dari proses ini adalah susunan populasi baru yang telah dirubah sesuai dengan parameter yang dimasukkan.

1	public ES(String nameFile, int maxPopulasi, double offspring, int iteration) {
2	file = new Array(nameFile);
3	this.maxPopulasi = maxPopulasi;
4	maxGen=(int) (offspring * maxPopulasi);
5	isi = new double[maxPopulasi + maxPopulasi * maxGen][file.isi.length][file.isi[0].length];
6	populasi(this.maxPopulasi);
7	variansi = variansi();
8	random1 = random(maxGen);
9	random2 = random(maxGen);
10	int stoper = 0;

Kode 5.4 Proses inisialisasi *evolution strategies*

Kode 5.4 merupakan proses inisialisasi dalam *evolution strategies*.
Implementasi Proses Inisialisasi

11. Baris nomor 1 terdapat kepala method dengan parameter parameter yaitu data asli, jumlah populasi, jumlah offspring, dan jumlah generasi.
12. Baris nomor 2 berisi proses input data dari file ke array.
13. Baris nomor 4-5 berisi proses perhitungan inputan yang berupa jumlah populasi dan jumlah generasi.
14. Baris nomor 7 adalah proses untuk membuat variansi berdasarkan data input.
15. Baris nomor 8-9 adalah proses untuk membuat susunan gen secara acak sesuai nilai parameter yang dilakukan sebanyak jumlah data untuk setiap populasi.
16. Baris ke 10 berisi kondisi berhenti.

Proses ini bertujuan membentuk populasi sesuai dengan parameter yang dimasukkan untuk diolah menggunakan *evolution strategies*. Proses ini dijalankan sebelum *evolution strategies* dengan hasil berupa array 2 dimensi yang memuat susunan gen tiap individu dalam populasi.

5.4.1 Implementasi Inisialisasi Populasi

Poses ini adalah tahapan salah satu tahapan pertama yang dilakukan dalam siklus algoritma *Evolution strategies*. Proses inisialisasi populasi awal dilakukan dengan cara mengacak nilai setiap gen yang membentuk kromosom. Jumlah populasi awal ditentukan oleh user saat parameter diisi pada halaman antarmuka. Proses implementasi inisialisasi populasi awal akan di jabarkan pada Kode Program 5.5 berikut.



1	public void populasi(int maxPopulasi) {
2	this.populasi = new double[maxPopulasi][file.isi.length];
3	for (int i = 0; i < populasi.length; i++) {
4	for (int j = 0; j < populasi[0].length; j++) {
5	populasi[i][j] = Math.random();
6	}
7	}
8	}

Kode 5.5 Proses Inisialisasi Populasi awal

Kode 5.7 bertujuan untuk saling menukar susunan gen pada induk terpilih. Berikut penjelasan langkah-langkah untuk kode tersebut.

1. Baris nomor 2 terdapat deklarasi variable array dengan tipe double untuk tempat penyimpanan nilai setiap populasi.
2. Baris nomor 3-5 terdapat kondisi perulangan proses random untuk setiap populasi.

5.4.2 Implementasi Perhitungan Generasi

Proses ini bertujuan untuk menghasilkan individu baru yang mirip dengan induk terpilih tapi berbeda satu gen. Hasil akhir dari proses ini akan ditambahkan dalam *array* populasi.

1	public double[][] gen(int check) {
2	double[][] gen = new double[N.length][file.isi.length];
3	System.out.println("gen");
4	int rule = 1;
5	int checkpoint = 0;
6	for (int i = 0; i < N.length; i++) {
7	if (i == rule * check - 1) {
8	rule++;
9	checkpoint = 0;
10	}
11	System.out.println("gen" + " " + i);
12	for (int j = 0; j < N[0].length; j++) {
13	gen[i][j] = populasi[checkpoint][j] + variansi[checkpoint][j] * N[i][j];
14	System.out.print(gen[i][j] + " ");
15	
16	}
17	System.out.println("");
18	checkpoint++;

19	}
20	return gen;
21	}

Kode 5.6 Proses Perhitungan Gen

Kode 5.6 merupakan proses untuk menghasilkan gen. Berikut penjelasan langkah-langkah untuk kode tersebut.

1. Baris nomor 1 terdapat kepala *method* dengan parameter berupa populasi yang akan digunakan dalam proses mutasi.
2. Baris nomor 2 terdapat proses deklarasi array dengan tipe double.
3. Baris nomor 6-10 terdapat proses perulangan untuk menentukan gen urutan ke i.
4. Baris nomor 12-16 terdapat proses perulangan untuk menghitung nilai gen berdasarkan parameter.

5.5 Implementasi Tampilan Antarmuka

Subbab ini membahas tampilan antarmuka sistem beserta fungsinya, dimana akan dijelaskan secara mendetail bagian-bagian dari tampilan antar muka pengolahan data pada aplikasi.

Gambar 5. 1 Tampilan awal antarmuka pengolahan data

Gambar 5.1 merupakan tampilan antarmuka pengolahan data sebelum data yang aktual di inputkan. Bagian atas dari tampilan ini memuat kolom untuk

memasukkan data serta parameter-parameter yang akan digunakan dalam proses optimasi.

The screenshot shows the ES software window with the following details:

- Data:** F:\r\Linier\Linier\data.txt (with a Browse button)
- Populasi:** 30
- Offspring:** 0.5
- Iteration:** 10
- ES Tab:** Active, showing sub-tabs: Data, Populasi, Variasi, Random 1, Random 2, Data Akhir, Fitness Akhir.
- Data Table:**

No	x(Kelemba...	y(Volume ...
1	3.75	1.325
2	4.87	2.1
3	4.16	1.4125
4	1.97	0.325
5	1.54	0.1625
6	1.47	0.1
7	3.98	1.65
8	3.47	1.2625
9	2.52	0.4625
10	3.87	1.3625
11	4.43	1.7
12	3.28	1.1625
13	2.11	0.5125
14	3.51	1.3
- Submit:** A button at the bottom left.

Gambar 5. 2Tampilan antarmuka pengolahan data

Gambar 5.2 merupakan tampilan antarmuka pengolahan data setelah data dan parameter-parameter optimasi di proses.

The screenshot shows the ES software window with the following details:

- Data:** F:\r\Linier\Linier\data.txt (with a Browse button)
- Populasi:** 30
- Offspring:** 0.5
- Iteration:** 10
- ES Tab:** Active, showing sub-tabs: Data, Populasi, Variasi, Random 1, Random 2, Data Akhir, Fitness Akhir.
- Populasi Table:**

No	Populasi 1	Populasi 2	Populasi 3	Populasi 4	Populasi 5	Populasi 6	Populasi 7	Populasi 8	Populasi 9	Populasi ...
1	0.935841...	0.338037...	0.223047...	0.966177...	0.688343...	0.037565...	0.648175...	0.787563...	0.585064...	0.543341...
2	0.813800...	0.632222...	0.984099...	0.161004...	0.620671...	0.171179...	0.168046...	0.172571...	0.876301...	0.199938...
3	0.692555...	0.659478...	0.246080...	0.766968...	0.827405...	0.392980...	0.688904...	0.938455...	0.590782...	0.627466...
4	0.415936...	0.484501...	0.464961...	0.339987...	0.349675...	0.471623...	0.501424...	0.806270...	0.085740...	0.036917...
5	0.507306...	0.197179...	0.502633...	0.975182...	0.445610...	0.384314...	0.127778...	0.475809...	0.411718...	0.935678...
6	0.531252...	0.833797...	0.824695...	0.932854...	0.812905...	0.034235...	0.534045...	0.788862...	0.463912...	0.586281...
7	0.602680...	0.562500...	0.204218...	0.682226...	0.861419...	0.747157...	0.084342...	0.555296...	0.244205...	0.923252...
8	0.290471...	0.387398...	0.293170...	0.714174...	0.567300...	0.526773...	0.207261...	0.137279...	0.798509...	0.129625...
9	0.361787...	0.618501...	0.361595...	0.402393...	0.123047...	0.924559...	0.005523...	0.677124...	0.142946...	0.947628...
10	0.664949...	0.632879...	0.872482...	0.424546...	0.158383...	0.125572...	0.047833...	0.376036...	0.031065...	0.311850...
11	0.396141...	0.966165...	0.627939...	0.694261...	0.093625...	0.572949...	0.014520...	0.043977...	0.997556...	0.310484...
12	0.152083...	0.019830...	0.682281...	0.601575...	0.720718...	0.304320...	0.324210...	0.601524...	0.668557...	0.610577...
13	0.573598...	0.942553...	0.434171...	0.360627...	0.329803...	0.603096...	0.118164...	0.726159...	0.703337...	0.933140...
- Submit:** A button at the bottom left.

Gambar 5. 3Tampilan antarmuka pengolahan data populasi

Gambar 5.3 merupakan tampilan hasil perhitungan populasi dari parameter-parameter optimasi yang telah di inputkan.

ES

</

Gambar 5. 4Tampilan antarmuka pengolahan data variansi

Gambar 5.4 merupakan tampilan hasil perhitungan variansi berdasarkan perhitungan populasi dari parameter-parameter optimasi yang telah di inputkan

ES

Data

F:\Linier\Linier\data.txt

Browse

Populasi

30

Offspring

0.5

Iteration

10

ES

Data	Populasi	Variansi	Random 1	Random 2	Data Akhir	Fitness Akhir
------	----------	----------	----------	----------	------------	---------------

No	Random 1	Random 2	Random 3	Random 4	Random 5	Random 6	Random 7	Random 8	Random 9	Random 10
1	0.487578...	0.676183...	0.687267...	0.493286...	0.664362...	0.791597...	0.308195...	0.029566...	0.779306...	0.737734...
2	0.237560...	0.115462...	0.708978...	0.373413...	0.493241...	0.911614...	0.172842...	0.327104...	0.836170...	0.115508...
3	0.130948...	0.289464...	0.870128...	0.632187...	0.550247...	0.812346...	0.464809...	0.079156...	0.563636...	0.297576...
4	0.693889...	0.552534...	0.413161...	0.132333...	0.144835...	0.274371...	0.756592...	0.685281...	0.047458...	0.440634...
5	0.047459...	0.618351...	0.954164...	0.928687...	0.167570...	0.198578...	0.935228...	0.256619...	0.211910...	0.295895...
6	0.033778...	0.784713...	0.879349...	0.562642...	0.913012...	0.293877...	0.349536...	0.248684...	0.400695...	0.575015...
7	0.219263...	0.202962...	0.892272...	0.011825...	0.788976...	0.407856...	0.616939...	0.810753...	0.383558...	0.288668...
8	0.896768...	0.142102...	0.179195...	0.493069...	0.292351...	0.208358...	0.351236...	0.975172...	0.972886...	0.692365...
9	0.918208...	0.969136...	0.472748...	0.168078...	0.364597...	0.180346...	0.371412...	0.749729...	0.022639...	0.578735...
10	0.313871...	0.604306...	0.485301...	0.624324...	0.987312...	0.383640...	0.322151...	0.890634...	0.471396...	0.131462...
11	0.448801...	0.775696...	0.836409...	0.014611...	0.801018...	0.541999...	0.568597...	0.494729...	0.349657...	0.978465...
12	0.857270...	0.301410...	0.218072...	0.449913...	0.961528...	0.268532...	0.132070...	0.226232...	0.037936...	0.572872...
13	0.731880...	0.475899...	0.304221...	0.711332...	0.142877...	0.339166...	0.893026...	0.701435...	0.197823...	0.508781...

Submit

Gambar 5. 5 Tampilan antarmuka pengolahan data random 1

Gambar 5.5 merupakan tampilan hasil proses random ke 1 berdasarkan parameter parameter yang telah di inputkan.

ES

Data

F:\rLinier\rLinier\data.txt

Browse

Populasi

30

Offspring

0.5

Iteration

10

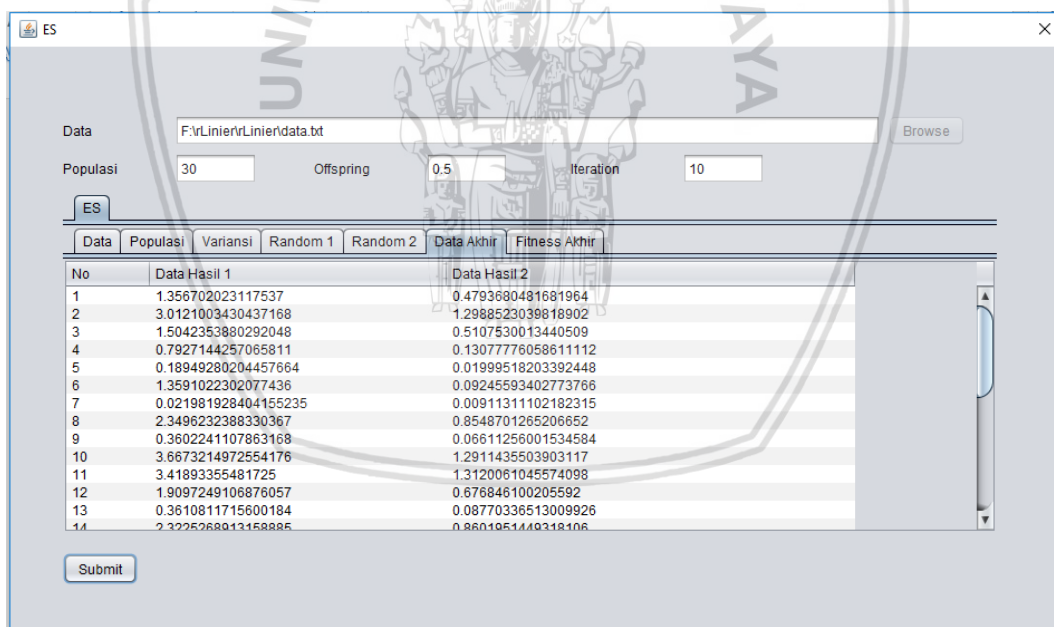
ES

Data	Populasi	Variansi	Random 1	Random 2	Data Akhir	Fitness Akhir				
No	Random 1	Random 2	Random 3	Random 4	Random 5	Random 6	Random 7	Random 8	Random 9	Random 10
1	0.767999...	0.380613...	0.021651...	0.969832...	0.098776...	0.599174...	0.520232...	0.535222...	0.826270...	0.124032...
2	0.953349...	0.945292...	0.246908...	0.349018...	0.577351...	0.444990...	0.648217...	0.740277...	0.854948...	0.876636...
3	0.403739...	0.522824...	0.256889...	0.248395...	0.058575...	0.817731...	0.543865...	0.655334...	0.322691...	0.196048...
4	0.869223...	0.190290...	0.170403...	0.837631...	0.257147...	0.478114...	0.818230...	0.091012...	0.247531...	0.509134...
5	0.188500...	0.737539...	0.141104...	0.737749...	0.181861...	0.136160...	0.536232...	0.167779...	0.817268...	0.456938...
6	0.131045...	0.508850...	0.647681...	0.489087...	0.285487...	0.022643...	0.475122...	0.769206...	0.431213...	0.826902...
7	0.092992...	0.974414...	0.995898...	0.540298...	0.930211...	0.862224...	0.066676...	0.128472...	0.550839...	0.693815...
8	0.008008...	0.193446...	0.701900...	0.295756...	0.707467...	0.807839...	0.800257...	0.076377...	0.723050...	0.559625...
9	0.710596...	0.509701...	0.473666...	0.195349...	0.780821...	0.259117...	0.815229...	0.891024...	0.730132...	0.567939...
10	0.149437...	0.524778...	0.208927...	0.125141...	0.246308...	0.147911...	0.022447...	0.604240...	0.206390...	0.410775...
11	0.140939...	0.231179...	0.960030...	0.729390...	0.281240...	0.993984...	0.948631...	0.753763...	0.904156...	0.261838...
12	0.798854...	0.809035...	0.277823...	0.270208...	0.006787...	0.123295...	0.599244...	0.935697...	0.924234...	0.928171...
13	0.816816...	0.947862...	0.008077...	0.464258...	0.870128...	0.006399...	0.661845...	0.668508...	0.902901...	0.672992...

Submit

Gambar 5. 6 Tampilan antarmuka pengolahan data random 2

Gambar 5.6 merupakan tampilan hasil proses random ke 2 berdasarkan parameter parameter yang telah di inputkan.



Data: F:\rLinier\rLinier\data.txt

Populasi: 30 Offspring: 0.5 Iteration: 10

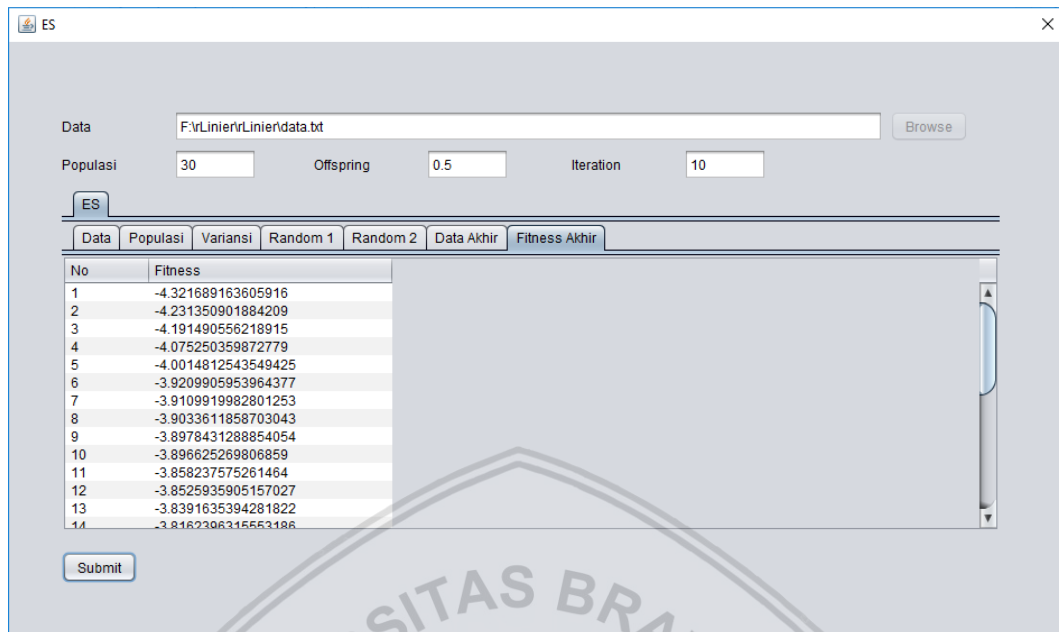
ES

Data	Populasi	Variansi	Random 1	Random 2	Data Akhir	Fitness Akhir
No	Data Hasil 1				Data Hasil 2	
1	1.356702023117537				0.4793680481681964	
2	3.0121003430437168				1.2988523039818902	
3	1.5042353880292048				0.5107530013440509	
4	0.7927144257065811				0.13077776058611112	
5	0.18949280204457664				0.01999518203392448	
6	1.3591022302077436				0.09245593402773766	
7	0.021981928404155235				0.00911311102182315	
8	2.3496232388330367				0.8548701265206652	
9	0.3602241107863168				0.06611256001534584	
10	3.6673214972554176				1.2911435503903117	
11	3.41893355481725				1.3120061045574098	
12	1.9097249106876057				0.676846100205592	
13	0.3610811715600184				0.08770336513009926	
14	2.3225268913158885				0.861051410318106	

Submit

Gambar 5. 7 Tampilan antarmuka pengolahan data akhir

Gambar 5.7 merupakan tampilan hasil perhitungan data akhir berdasarkan hasil dari proses random ke 1 dan 2 berdasarkan parameter parameter yang telah di inputkan.



Gambar 5. 8Tampilan antarmuka pengolahan data fitness akhir

Gambar 5.8 merupakan tampilan hasil fitness akhir dari perhitungan data akhir.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini menjelaskan hasil pengujian dan analisis terhadap implementasi sistem untuk membuktikan kebenaran metode dari kajian pustaka yang dibahas pada bab sebelumnya. Pada penelitian ini pengujian akan dibagi menjadi 2 bagian yaitu pengujian parameter *evolution strategies* serta pengujian kekuatan persamaan regresi dengan dan tanpa optimasi. Pada setiap bagian, analisis akan dilakukan untuk menarik kesimpulan hasil pengujian.

6.1 Pengujian Parameter *Evolution strategies*

Pengujian parameter *evolution strategies* dilakukan untuk mencari parameter yang dapat menghasilkan *fitness* terbaik secara efektif dan efisien. Pengujian parameter *evolution strategies* dilakukan dengan mengambil nilai *fitness* terbaik secara berurutan dimulai dari variasi jumlah individu, jumlah *offspring* serta jumlah generasi. Pengujian parameter *evolution strategies* dilakukan masing-masing 10 kali kemudian diambil yang terbaik, hal ini dilakukan karena hasil yang diberikan *evolution strategies* akan berbeda tiap kali dijalankan.

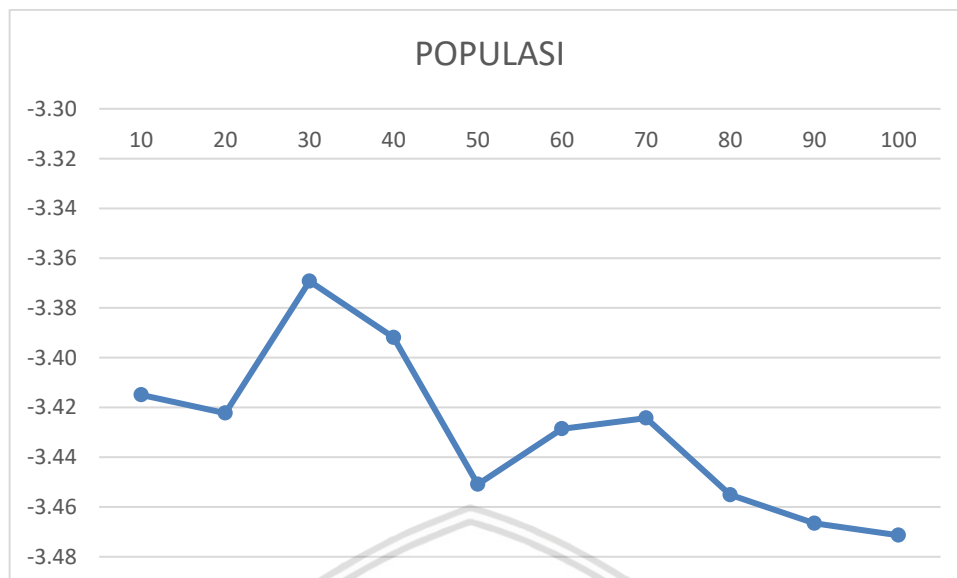
6.1.1 Pengujian Jumlah Populasi

Pengujian ini bertujuan mencari jumlah populasi yang dapat menghasilkan *fitness* terbaik secara efektif dan efisien. Pengujian ini dilakukan pada jumlah populasi 10 hingga 100 dengan kelipatan 10. Parameter lain yang digunakan pada pengujian ini yaitu jumlah generasi 10, jumlah *offspring* sebesar 0,5 μ . Hasil pengujian jumlah individu ditunjukkan pada Tabel 6.1.

Table 6.1 Hasil pengujian jumlah populasi

Jumlah Populasi	Fitness percobaan ke-										rata-rata fitness
	1	2	3	4	5	6	7	8	9	10	
10	-3.53	-3.21	-3.35	-3.59	-3.44	-3.45	-3.28	-3.33	-3.42	-3.55	-3.41
20	-3.39	-3.37	-3.44	-3.41	-3.49	-3.52	-3.36	-3.31	-3.49	-3.44	-3.42
30	-3.40	-3.27	-3.33	-3.40	-3.30	-3.43	-3.44	-3.32	-3.49	-3.30	-3.37
40	-3.41	-3.48	-3.47	-3.32	-3.31	-3.45	-3.31	-3.37	-3.33	-3.48	-3.39
50	-3.44	-3.44	-3.49	-3.46	-3.41	-3.40	-3.47	-3.54	-3.49	-3.38	-3.45
60	-3.39	-3.48	-3.40	-3.39	-3.45	-3.40	-3.37	-3.53	-3.46	-3.40	-3.43
70	-3.37	-3.35	-3.40	-3.44	-3.47	-3.43	-3.48	-3.44	-3.44	-3.43	-3.42
80	-3.50	-3.47	-3.47	-3.43	-3.47	-3.39	-3.44	-3.44	-3.48	-3.45	-3.46
90	-3.43	-3.49	-3.39	-3.47	-3.54	-3.47	-3.43	-3.45	-3.49	-3.50	-3.47
100	-3.48	-3.45	-3.46	-3.47	-3.48	-3.50	-3.56	-3.43	-3.42	-3.46	-3.47

Tabel 6.1 merupakan hasil pengujian jumlah populasi terbaik untuk digunakan. Dari tabel tersebut dapat dibuat sebuah grafik yang diperoleh dari nilai rata-rata *fitness* untuk mempermudah dalam membaca hasil pengujian. Grafik hasil pengujian jumlah populasi terbaik ditunjukkan pada Gambar 6.1.



Gambar 6.1 Grafik pengujian jumlah individu

Gambar 6.1 merupakan grafik hasil pengujian jumlah individu terbaik untuk digunakan. Grafik tersebut menunjukkan kecenderungan *fitness* yang semakin baik seiring dengan bertambahnya jumlah individu, namun hasil terbaik diperoleh dari jumlah individu 30. Hal ini terjadi karena semakin banyak jumlah individu kemungkinan individu terbaik terpilih sebagai induk akan semakin kecil. Dari hasil pengujian dapat disimpulkan bahwa secara garis besar semakin banyak jumlah individu belum tentu perolehan *fitness* semakin baik, jika terlalu banyak tidak akan menghasilkan peningkatan yang berarti bahkan bisa menurun.

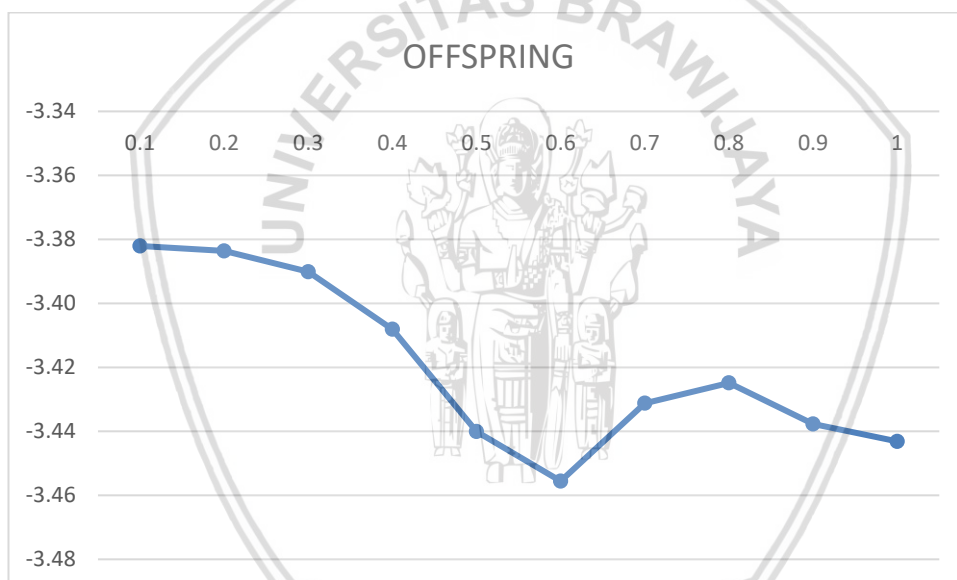
6.1.2 Pengujian Jumlah Offspring

Pengujian ini bertujuan mencari jumlah populasi yang dapat menghasilkan *fitness* terbaik secara efektif dan efisien. Pengujian ini dilakukan pada jumlah *offspring* 0,1 μ hingga 1 μ dengan kelipatan 0,1. Parameter lain yang digunakan pada pengujian ini yaitu jumlah generasi 10, jumlah populasi sebesar 40. Hasil pengujian jumlah individu ditunjukkan pada Tabel 6.2

Table 6.2 Hasil Pengujian Jumlah *Offspring*

Jumlah Offspring	Fitness percobaan ke-										rata-rata fitness
	1	2	3	4	5	6	7	8	9	10	
0.1	-3.36	-3.35	-3.40	-3.42	-3.32	-3.33	-3.46	-3.34	-3.36	-3.47	-3.38
0.2	-3.37	-3.45	-3.38	-3.34	-3.46	-3.36	-3.32	-3.37	-3.33	-3.44	-3.38
0.3	-3.35	-3.37	-3.51	-3.42	-3.45	-3.26	-3.48	-3.39	-3.33	-3.34	-3.39
0.4	-3.36	-3.31	-3.49	-3.53	-3.37	-3.48	-3.32	-3.29	-3.57	-3.37	-3.41
0.5	-3.37	-3.37	-3.54	-3.51	-3.49	-3.50	-3.44	-3.44	-3.40	-3.34	-3.44
0.6	-3.46	-3.39	-3.52	-3.44	-3.48	-3.43	-3.47	-3.38	-3.48	-3.51	-3.46
0.7	-3.49	-3.52	-3.42	-3.39	-3.36	-3.50	-3.37	-3.37	-3.48	-3.42	-3.43
0.8	-3.40	-3.52	-3.40	-3.42	-3.42	-3.45	-3.44	-3.43	-3.45	-3.33	-3.42
0.9	-3.49	-3.43	-3.41	-3.54	-3.51	-3.36	-3.29	-3.42	-3.38	-3.55	-3.44
1	-3.44	-3.51	-3.40	-3.43	-3.36	-3.48	-3.47	-3.45	-3.43	-3.46	-3.44

Tabel 6.2 merupakan hasil pengujian jumlah *offspring* terbaik untuk digunakan. Dari tabel tersebut dapat dibuat sebuah grafik yang diperoleh dari nilai rata-rata *fitness* untuk mempermudah dalam membaca hasil pengujian. Grafik hasil pengujian jumlah *offspring* terbaik ditunjukkan pada Gambar 6.2

Gambar 6.2 Grafik Pengujian Jumlah *Offspring*

Hasil pengujian menunjukkan bahwa nilai rata-rata fitness tertinggi pada saat jumlah *offspring* 0.1. Gambar 6.2. menjelaskan bahwa tidak selamanya semakin besar jumlah *offspring* yang dihasilkan maka akan menghasilkan nilai fitness yang lebih baik. Semakin banyak *offspring* yang dihasilkan maka akan menghasilkan variasi kromosom yang lebih beragam. Namun, banyaknya variasi kromosom yang dibentuk tidak menjamin hasil yang lebih optimal. Banyaknya variasi kromosom yang terbentuk hanya akan memperbesar peluang terjadinya hasil yang lebih optimal. Pada pengujian ini dapat dikatakan bahwa semakin sedikit jumlah *offspring* yang dihasilkan maka semakin sedikit variasi kromosom yang akan

dihasilkan sehingga sedikitnya peluang untuk mencapai hasil yang optimal, dan begitu juga sebaliknya.

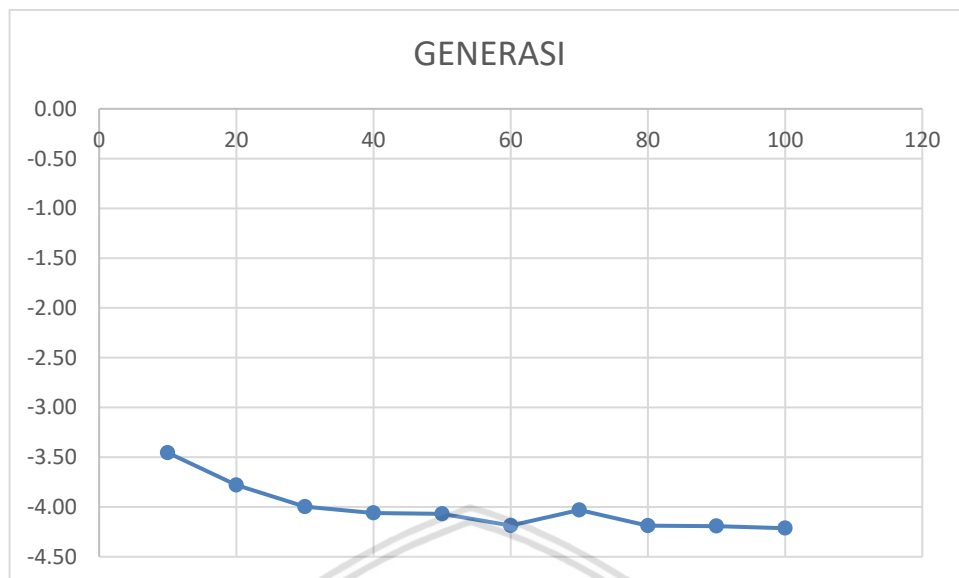
6.1.3 Pengujian Jumlah Generasi

Pengujian ini bertujuan mencari jumlah generasi yang dapat menghasilkan fitness terbaik secara efektif dan efisien. Pengujian ini dilakukan pada jumlah generasi 10 hingga 100 dengan kelipatan 10. Parameter lain yang digunakan pada pengujian ini yaitu jumlah populasi 40, jumlah *offspring* 0.5 μ . Hasil pengujian jumlah generasi pada penelitian ini ditunjukkan pada Tabel 6.2.

Table 6.3 Hasil Pengujian Jumlah Generasi

Jumlah Generasi	Fitness percobaan ke-										rata-rata fitness
	1	2	3	4	5	6	7	8	9	10	
10	-3.59	-3.42	-3.43	-3.48	-3.51	-3.38	-3.48	-3.48	-3.44	-3.33	-3.45
20	-3.75	-3.80	-3.74	-3.83	-3.84	-3.78	-3.72	-3.88	-3.75	-3.72	-3.78
30	-4.05	-3.98	-3.91	-4.16	-4.03	-4.16	-3.83	-3.93	-3.90	-4.01	-4.00
40	-4.00	-4.49	-4.08	-3.97	-3.90	-3.83	-4.07	-3.93	-4.23	-4.10	-4.06
50	-4.19	-4.08	-4.08	-3.96	-4.10	-4.07	-4.01	-3.87	-4.35	-3.99	-4.07
60	-4.22	-3.97	-4.71	-4.07	-4.07	-3.96	-4.53	-4.02	-4.03	-4.27	-4.19
70	-3.92	-4.07	-3.97	-4.12	-3.96	-4.06	-4.21	-3.99	-3.97	-4.04	-4.03
80	-4.75	-4.34	-4.09	-4.12	-4.25	-4.06	-4.10	-4.05	-4.14	-3.99	-4.19
90	-4.21	-4.12	-4.45	-4.21	-4.02	-4.01	-4.20	-4.11	-4.37	-4.24	-4.19
100	-4.26	-4.30	-4.29	-4.29	-3.92	-4.13	-4.15	-4.29	-4.39	-4.10	-4.21

Tabel 6.3 merupakan hasil pengujian jumlah generasi yang dapat menghasilkan nilai *fitness* terbaik. Dari tabel tersebut dapat dibuat sebuah grafik yang diperoleh dari nilai rata-rata *fitness* untuk mempermudah dalam membaca hasil pengujian. Grafik hasil pengujian jumlah generasi terbaik ditunjukkan pada Gambar 6.3.



Gambar 6.3 Grafik Pengujian Generasi

Gambar 6.3 merupakan grafik hasil pengujian jumlah generasi. Grafik tersebut menunjukkan kecenderungan perolehan *fitness* yang semakin menurun dengan bertambahnya jumlah generasi, dan tidak ada perubahan *fitness* yang signifikan semakin bertambahnya jumlah generasi. Dari hasil analisis di atas dapat ditarik kesimpulan bahwa jumlah generasi yang baik merupakan jumlah generasi terbaik yang dapat menghasilkan *fitness* terbaik.

BAB 7 PENUTUP

Bab ini merupakan bab terakhir yang membahas kesimpulan beserta saran untuk penelitian ini. Bab ini akan menjelaskan hasil dari penelitian yang sudah disimpulkan berdasarkan rumusan masalah. Selain hasil penelitian, bab ini juga akan membahas tentang saran untuk memperbaiki kekurangan penelitian ini guna menjadi masukan pada penelitian berikutnya.

7.1 Kesimpulan

Kesimpulan pada penelitian ini dibuat sebagai jawaban atas rumusan masalah yang dibuktikan dengan hasil pengujian. Berdasarkan rumusan masalah dan hasil pengujian yang sudah dilakukan pada bab sebelumnya dapat ditarik kesimpulan sebagai berikut.

Regresi linier sebagai alat prediksi dapat dioptimasi menggunakan *evolution strategies* dengan nilai kriteria informasi sebagai *fitness*, dengan syarat parameter *evolution strategies* yang digunakan sesuai. Parameter yang sesuai untuk digunakan yaitu jumlah individu dan jumlah generasi, namun jika terlalu besar tidak akan ada kenaikan *fitness* bahkan bisa terjadi penurunan dengan semakin banyaknya proses. Pada penelitian ini *fitness* yang baik diperoleh dengan jumlah populasi 30 dan jumlah generasi 10. Parameter lainnya yaitu jumlah *offspring*, parameter ini dapat menghasilkan *fitness* yang baik jika cenderung seimbang. Pada penelitian ini jumlah *offspring* dengan *fitness* terbaik diperoleh pada 0.1 μ . Dan hasil *fitness* terbaik yang didapatkan adalah -3,47.

7.2 Saran

Saran pada penelitian ini dibuat sebagai masukan untuk penelitian berikutnya guna memperbaiki kekurangan pada penelitian ini. Berikut saran yang dapat dijadikan masukan untuk penelitian berikutnya.

Regresi linier dioptimasi dengan *evolution strategies* untuk mendeteksi amatan pencilan yang diterapkan pada penelitian ini dapat digunakan dengan beberapa variabel bebas, meskipun cara mencari persamaan regresi berbeda untuk setiap jumlah variabel bebas. Hal ini dikarenakan cara memperoleh *fitness* dan nilai kekuatan persamaan regresi tetap sama yaitu berdasarkan variabel terikat, mengingat regresi linier hanya memiliki 1 variabel terikat meskipun terdiri dari beberapa variabel bebas. Pada penelitian berikutnya variabel bebas dapat ditambahkan untuk menganalisis faktor lain yang mempengaruhi kebutuhan volume air selain kelembapan tanah.

DAFTAR PUSTAKA

- Alma, Ö. G., Kurt, S., & Ugur, A. (2008). Genetic algorithms for outlier detection in multiple regression with different information criteria. *Journal of Statistical Computation and Simulation*, 29–47.
- Balai Pengkajian Teknologi Pertanian Jawa Timur. (2010, 5 10). *Laboratorium Benih*. Retrieved 4 1, 2017, from Balai Pengkajian Teknologi Pertanian Jawa Timur: <http://jatim.litbang.pertanian.go.id/ind/index.php/layanan/laboratorium-benih>
- Draper, N. R., & Smith, H. (1998). *Analisis Regresi Terapan*. Jakarta: PT. Gramedia Pustaka Utama.
- FAO Land & Water Division. (2003, 3 1). *Crop Water Information*. Retrieved 1 4, 2017, from FAO Land & Water Management: <http://www.fao.org/nr/water/cropinfo.html>
- Gen, M., & Cheng, R. (1997). *Genetic Algorithms and Engineering Design*. New York: John Willey & Sons, Inc.
- Hadi, A. S., & Simonoff, J. S. (1993). Procedures for the Identification of Multiple Outliers in Linear Models. *Journal of the American Statistical Association*, 88.
- Johnson, R. A., & Wichern, D. W. (1996). *Applied Multivariate Statistical*. New Jersey: Prentice Hall of India Private Limited.
- Kenney, J. F., & Keeping, E. S. (1962). *Linear Regression and Correlation*. Princeton, NJ: Van Nostrand.
- Levin, R. I., & Rubin, D. S. (1998). *Statistics for management*. New delhi: Prentice Hall.
- Mahmudy, W. F. (2013). *Algoritma Evolusi*. Malang: Program Teknologi Informasi dan Ilmu Komputer (PTIIK) Universitas Brawijaya.
- Mills, J., & Prasad, K. (1992). A Comparison of Model Selection Criteria. *Econometric Rev* 11, 201–233.
- R Development Core Team. (2008). *A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing.
- Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2), 461-464.
- Software and Systems Engineering Vocabulary*. (2008, 7 31). Retrieved 11 2016, 24, from SEVOCAB:

https://pascal.computer.org/sev_display/search.action;jsessionid=5f80746ac7214cf2e2ea49dd3494

Tolvi, J. (2004). Genetic algorithms for outlier detection and variable selection in linear regression models. *Soft Computing* 8, 527–533.

Yansari, M., Ratnawati, D. E. & M., 2016. *Optimasi Biaya dan Asupan Gizi Pasien Diet Khusus Dengan Menggunakan Algoritma Evolution strategies*. Malang: Fakultas Ilmu Komputer Universitas Brawijaya.

